PITCH DETERMINATION OF SPEECH SIGNALS

by

Mark David Anderson

Submitted in partial fulfillment
of the requirements
for the degrees of

Master of Science in Electrical Engineering and Computer Science
and
Bachelor of Science in Electrical Engineering and Computer Science

at the
Massachusetts Institute of Technology

February, 1986

©Mark David Anderson 1986

The author hereby grants to MIT permission to reproduce and to distribute copies of this
thesis document in whole or in part.

Signature of Author

Department of Electrical Engineering and Computer Science

Certified by

Victor W. Zue
Thesis Supervisor (MIT)

Certified by

Janet Breckenridge Pierrehumbert
Company Supervisor (AT&T Bell Laboratories)

Accepted by

Arthur C. Smith
Chairman, Departmental Committee on Graduate Students
PITCH DETERMINATION OF SPEECH SIGNALS

by

Mark David Anderson

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of
Master of Science in Electrical Engineering and Computer Science
and Bachelor of Science in Electrical Engineering and Computer Science
February, 1986

ABSTRACT

Algorithms for the determination of the pitch of speech signals are a basic research tool in speech science, and a necessary part of many of the commercial applications of that science. The work described in this thesis concerns two distinct aspects of the problems of pitch determination. First, a study is made of preliminary issues, such as pitch definition, the relationship between pitch definitions and pitch determination algorithms, algorithm testing, and application areas. The foundation of this study is a typology of pitch definitions, using the categories of "production pitch," "mathematical pitch," and "reception pitch." Secondly, two new pitch determination algorithms are presented, which were motivated, developed, and evaluated on the basis of the ideas contained in the first part. One of these algorithms attempts to determine the locations of glottal closure from a speech signal; it does this by examining a processed form of the linear prediction residual. This algorithm was tested on an 80 sentence corpus against an electroglottographic standard, and was found to be very accurate and fairly reliable. The goal of the second algorithm is to find the time-varying fundamental of a quasi-periodic signal. This algorithm works by first finding the harmonics in a short-term spectral representation of the signal, and then determining the quasi-fundamental from those harmonics using a generalized greatest common divisor function. The results of limited testing of this algorithm are promising.

Thesis Supervisor: Dr. Victor Zue

Title: Associate Professor of Electrical Engineering
Preface

My introduction to speech research and linguistics came about as a result of the co-op program in the Electrical Engineering and Computer Science Department at MIT. By the mutually blind assignment procedure, my first summer assignment was to the then Speech Analysis and Linguistics Department of Bell Laboratories (now the Speech Analysis and Artificial Intelligence Department of AT&T Bell Laboratories). The match proved to be serendipitous, and I returned to the same department for my second and third assignments as well. This thesis is the result of work done in the third and last assignment at the Labs, and on work done at MIT during the term following that assignment.

I owe my biggest debts to three groups of individuals. First of all I owe a great debt to my parents, who gave—and are giving—so much of themselves to their children. Next, I owe something to Dr. Arnold E. Ross, whose unique summer program has for so many years encouraged the development of young mathematicians, and from whom I first learned to “think deeply of simple things.” Lastly, I owe a great deal to Dr. Janet Pierrehumbert, my patient mentor and friend for all three assignments, through whose gentle guidance I have learned so much.

Thanks are also due for the many stimulating conversations I have had with the members of the speech research groups of both MIT and Bell Laboratories. Special thanks are due to Corine Bickley for proofreading the text.

While doing this work I was partially supported by an MIT RLE research assistantship, and by an NSF Graduate Fellowship, which I gratefully acknowledge.
2 Pitch Determination

2.1 A Production Pitch Tracker .............................................. 67
  2.1.1 Background .............................................................. 67
  2.1.2 The Stages of Processing ............................................ 70
    2.1.2.1 The Basic Transformation ..................................... 72
    2.1.2.2 Simplification ................................................... 82
    2.1.2.3 Peak Finding .................................................... 86
    2.1.2.4 Adjusting the Peak Locations ................................ 92
    2.1.2.5 Extrapolation ................................................... 94
  2.1.3 Tests ........................................................................... 95
    2.1.3.1 Data ..................................................................... 95
    2.1.3.2 Software .............................................................. 98
    2.1.3.3 Results ............................................................... 100

2.2 A Mathematical Pitch Tracker ........................................... 109
  2.2.1 Spectrum Analysis ...................................................... 110
  2.2.2 Finding Harmonics ...................................................... 112
  2.2.3 Combining Harmonics .................................................. 113

2.3 Summary and Conclusion .................................................. 119

A The Structure of a Pitch Tracker ......................................... 121
  A.1 Primitive Extraction ...................................................... 121
  A.2 Local Processing .......................................................... 123
  A.3 Context Constraints ...................................................... 123
  A.4 Approximating Constraints: Control Structures ............... 126
    A.4.1 Markov Models ....................................................... 127
    A.4.2 Full-Fledged Dynamic Programming ............................ 128
    A.4.3 Island-Driving ...................................................... 129
    A.4.4 Relaxation ............................................................. 131
    A.4.5 Parsing ................................................................. 132
  A.5 Integrating Information ................................................ 135

Bibliography ....................................................................... 138
Introduction

They that touch pitch will be defiled.

Much Ado About Nothing, III, iii, 61

There is certainly no lack of journal papers or academic theses on the topic of pitch tracking. Ever since it became reasonable to process signals digitally, the task and its attendant problems have received a great deal of attention; one recently compiled bibliography is over 100 pages long [46]. Even in 1968 this superabundance was noticeable:

Our only apology for adding to the already large list of methods for pitch extraction described in the literature is that the methods described here are based upon promising new ideas not previously reported. [109]

Since then, there have been many more journal articles on the subject, and it is a perennial topic of ICASSP and ASA meeting papers. The "already large list" is getting larger all the time, and there is no indication that this trend will soon change.

With this kind of background, it would seem likely that another thesis on the topic would soon, like most of its predecessors, amount to nothing more than another tombstone in a burgeoning algorithmic graveyard. This is not necessarily cause for dismay; it is to be expected that just as one piece of scientific research supplants some previous work, so it will later be supplanted itself by future work. However, even a cursory look at the literature on pitch detection reveals that it does not reflect this kind of steady, cumulative progress that you might expect to find, and that is found in some other research areas. The inhabitants of this particular graveyard were not victims of murder; they simply died of old age. The best pitch trackers today aren't significantly better than those of five, ten, or even fifteen years ago, and chances are roughly 50-50 whether the next paper on the topic will report on an algorithm with better or worse performance statistics than the last. Totally new ideas on the topic have become increasingly rare; the literature has been dominated by reports on minor variations of past ideas. Neither kind of paper seems to be getting us anywhere; it's apparently a choice between walking randomly or walking in circles.
It would seem that merely specifying yet another pitch tracking algorithm actually accomplishes very little at best. Instead, it has now come time to take stock of the situation and to decide whether this apparent stagnation has been caused by basic confusions, by lack of appropriate algorithmic techniques, by intrinsic intractability of the problem, or by some combination of these factors.

This thesis can be seen as having two main parts, which, although very closely related, could conceivably stand alone. In the interest of freeing myself from a morbid precedent, I have devoted a substantial portion of this thesis to an examination of more preliminary questions than simply “How can I get a pitch contour from a speech wave?”, such as:

- What is pitch? What is a pitch tracker?
- What distinguishes the various definitions of pitch? What consequences does this have for the various applications of pitch trackers?
- What place do such attributes as computational complexity, cost, robustness in the face of noise, etc., have in the evaluation of pitch trackers?
- What does it mean for a pitch tracker to be wrong?
- How should pitch trackers be tested?

If word count is an accurate representation of how effort is distributed, these questions, whose answers are crucial to any real progress in solving the problem, have not been explicitly studied a great deal, and it is through them that I hope to make a contribution. Those issues are covered in the first chapter.

The more customary aim of the second chapter is to present some new pitch determination algorithms; these were motivated, developed, and evaluated on the basis of the ideas contained in the first chapter.

Since the title may mislead some readers, I should make it clear at this point that my main concern here is not psychoacoustic pitch. It is a commonplace that there is at least one other entity besides “real” pitch which often makes it necessary to disambiguate the word; the goal of this work is to make our understanding of these other entities more precise, and to give some algorithms for obtaining them. Psychoacoustic pitch plays mostly a contrastive role here, and no algorithm for determining it is given. There is currently no very good name for these other kinds of “pitch”; I have stolen my title from Hess’s book ([46]), hoping that by taking advantage of a precedent I can somehow reduce my own culpability. Nevertheless, this project is not without interest to those working on psychoacoustic pitch, because it is not simply the case that these other ideas of pitch need to be relieved from the influence of psychoacoustic pitch. The influence is mutual; all of these ideas affect, inform, and confuse each other, and making any one of them more distinct and explicit naturally improves our understanding of the others as well.
Chapter 1

Preliminaries to Pitch Determination

1.1 A Typology of Pitch Definitions

The first step, of course, is to define pitch.

However, this is no simple task. Three factors conspire to obfuscate the question that it becomes a significant challenge to answer it properly, and various aspects of it will occupy us for most of this chapter. Terminology is one of these confounding factors. There are a whole host of terms which are connected to the idea of pitch: “pitch,” “periodicity,” “fundamental,” “f0,” “excitation,” “regularity,” and “intonation.” All of these words are ambiguous, redundant, vague, and inconsistently used. Often researchers become frustrated with this lack of precision, and so they coin yet more terms, such as “virtual pitch,” “periodicity pitch,” “voice pitch,” and “laryngeal frequency.” Another problem associated with terminology is that attempts to talk about different kinds of pitch are often foiled by the strong normative feelings some people have about how the particular word “pitch” should be used. The second of the factors arises from the prominent role that pitch determination algorithms often play in discussions of pitch; in these cases pitch is usually defined operationally. This is a convenient approach, but it gives implementations primacy over the entity they are trying to obtain. The third factor is similar to the second; the use of a pitch determination algorithm in a particular application often results in pitch being defined functionally, in terms of the role it plays in that application. These three factors—having to do with terminology, implementation, and application—will be investigated more thoroughly in sections 1.3.1, 1.3.2, and 1.3.3, respectively. Before that, however, I would like to present a proposal for possible pitch definitions and their organization, once the effects of these factors have been eliminated.
There are two kinds of definitional problems with regard to pitch. The first kind, which is where the three factors above do most of their damage, is rather broad, and doesn’t have so much to do with getting the details of a specific definition right, as with distinguishing it from a radically different definition. To combat this, I propose the following typology of pitch definitions, a principled categorization of all definitions of pitch, past and future, which facilitates the differentiation of valid definitions, and the recognition of pseudo definitions. The second kind of problem consists in trying to pin down a single idea and express a precise formulation of it; of course this kind of definitional problem occurs in nearly every research enterprise. I address this problem in the next few sections by specifying several explicit definitions of pitch.

In this typology of pitch definitions, there are three categories: production pitch, mathematical pitch, and reception pitch.¹ These are not definitions themselves, but are labels for classes of definitions, general rubrics that definitions fall under. I claim that these classes form a partition; I will be arguing as we proceed that there is no intersection between any two of them. Although in this thesis I will be concerned primarily with speech signals, this categorization incorporates all pitch definitions, not just those notions of pitch associated with speech communication.

Hopefully the discussion so far, which has been rather abstract, will become clearer as we proceed.

1.1.1 Production Pitch

When we say that something produces a particular pitch, we almost certainly do not mean that literally. For example, the human speech apparatus doesn’t produce pitch, it produces speech. When the word “pitch” is used to refer to a production mechanism, it refers (usually) to some rate parameter which describes that mechanism, or perhaps to an abstract parameter which controls repeating phenomena. I use the term production pitch to refer to a class of pitch definitions wherein the pitch of a signal is taken to be a characterization of the origin or generation of that signal. Thus this interpretation refers to a very special case of a very general situation: the problem of finding an appropriate characterization of the control, or at least of the origin, of a final output signal. Such a characterization can be found for any physical signal, whether or not it exhibits periodic behavior. For a stock index function it could be the values of various parameters representing market pressures; for the sound waves coming from a flute it could be some indication of the configuration of the player’s fingers, tongue, lips, and cheeks, along with the volume flow at the mouthpiece.

There are several different kinds of signals which superficially are somewhat periodic. The periodicity could have a direct correlate in the characterization; the rpm

¹While the invention of yet more terms in this area may seem gratuitous, in this case I think it is justified. These are, admittedly, not very pleasing expressions, but no alternatives without misleading connotations were apparent.
of an engine, which is a frequency measure, is directly related to the fuel mixture. The frequency of the whistle of a tea pot is directly related to the built up pressure and the aperture area. Or perhaps the periodicity could have no place whatsoever in the characterization; the flute model given above is an example. Or we could have an intermediary situation; clapping and human speech production could be considered to be examples of this case. The point is that for the task we are considering now our primary concern is not with whether the signal appears periodic, or if it does, with what that periodicity is; we are concerned simply with finding a suitable characterization of its generation. When a rate-like parameter would be appropriate in the characterization, it is part of the cause, the control, whereas any apparent periodicity in the final result is part of the effect.

Note that for production pitch, the (definition of) pitch of a signal will change according to what the source of the signal is, and even for a single source there might be several possible definitions, either from competing theories, or from different levels of description within a single theory.

Before proceeding to a discussion of how production pitch might be defined for the acoustic signals originating from human speech production, three points should be made. First of all, in defining the pitch of a speech signal we aren’t interested in everything that went into producing a speech signal, but only in what goes on at the larynx. This does not mean that we are assuming that the behavior of the articulators is independent of what goes on in the source, nor does it mean that we are assuming that we can somehow remove all the effects of the articulators without figuring out what they are doing—say by some inverse filtering process. It is just that no matter what additional information we may acquire in our pitch determination algorithm, all that we will “return” is the characterization of the source. Secondly, the choice between the different possible definitions is not arbitrary, though many valid ones are possible. Some considerations are that the description be obtainable, that there be easily found data on it, that it be informative, that it be useful for applications, and that it be meaningful for as many cases as possible. Two people weighting those factors differently might choose differently. Thirdly, the characterization chosen to define the pitch of some production mechanism is necessarily phrased in the context of some theory of that mechanism—e.g., different economic theories might have radically different analyses of the underlying processes producing a stock index. This means that a definition of production pitch for speech which invoked such entities as zero-crossing slopes or waveform extrema in the speech signal would be unacceptable. Those primitives might be used in an implementation because they provide a simple approximation to the ideal primitives, but certainly the definition of the goal of the implementation shouldn’t rely on those phenomena, since they have an extremely obscure and unsystematic relationship to the actions of the production mechanism.

There are at least three different kinds of definitions for the production pitch of speech; each of these succeeding characterizations corresponds to a higher and higher-level account of the production mechanism.

The first of these is at the lowest level; it is a simple external summary of what is
happening at the larynx. The description could, for example, give the glottal opening area, or the glottal acoustic flow, as a function of time. However, if one wants to determine the characterization from speech signals alone, then these leave something to be desired, because although it is possible that there is a one-to-one function from speech signals to glottal flow functions—or if there isn't, one could find the whole inverse image of a speech signal—experience has shown that finding this function is extremely difficult ([5]). Most attempts at this so far have involved manual selection of analysis frames, and then manual tuning of an inverse filter—and even then the authenticity of the result is no altogether certain. Thus we are motivated to find another low-level characterization which may not be as accurate, robust, or revealing as these, but is easier to determine. A description that is often chosen is a listing of the times of the instants of glottal closure. Clearly this gives less information than the previous suggestions, but it has the decided advantage that it is comparatively straightforward to collect relevant data on glottal closure times, if one has access to a throat accelerometer or an electroglottograph. Part of the problem with attempting to model glottal flow is that it is hard to get any dependable data on it. For determination of glottal area, the situation is slightly better, since stroboscopic photography and fibroscopes do give a fairly accurate picture of the area through time, but for both glottal flow and area the sheer quantity of data that it is necessary to process, and then explain, is staggering. Thus we conclude that glottal closure times are easier to obtain.

It is also true that for most researchers the glottal closure time characterization is more useful. As we shall see in Section 1.4.2, this characterization is very helpful in speech analysis, synthesis, recognition, and coding, and many of those applications would not benefit much from more detailed information. Even for those applications where this characterization is inadequate—for some very accurate speech analysis, or for voice pathology—glottal closure times are extremely useful for subsequent signal processing.

It is not true, however, that glottal closure times constitute as robust a characterization as the other low-level proposals—glottal area and glottal acoustic flow. Occasionally speakers will not have a complete closure or will have none at all; certainly for breathy or high-pitched voices there is often no full closure ([46]). The obvious solutions for this—for example, changing the definition of the characterization to be “time of minimum glottal area” or “time of maximum glottal excitation”—are not only sketchy, but at best they sometimes will still not tell be very informative. Another problem with the characterization is that the glottal folds are not just two points which are either together or apart; nor are they one or even two dimensional objects. They are three dimensional, and the event that we refer to as simply closure is actually complicated; the contact area function through time may not be monotonic through one opening or closing cycle, and the portions of the folds that touch may not be continuous. Thus the notion that there is a single time which can be called the closure time is only a very rough approximation.

A lot of information is lost in a pitch characterization using just glottal closure times. For example, it is possible that there are qualitatively different kinds of closures
which are being collapsed into one entity by this procedure. Moreover, there are other laryngeal events that are also of importance both for speech analysis and for signal processing; in particular, this doesn’t help in the effort to find glottal opening times except in the obvious sense that exactly one opening occurs between two closures (whatever one’s definitions, they should satisfy that). However, despite the fact that some glottographs provide data on glottal opening times as easily, or more easily, than they do on closure times, it is a fact that openings usually have a much less pronounced effect on the speech waveform, so that they are both harder to find and not as useful if they are found.

In Section 1.2.3 I will describe how this glottal closure location representation may be converted into a pitch contour (rate as a function of time). The pitch contours resulting from this transformation will reflect, with the proper rises and falls, the micromelody: the perturbations in an overall regular sequence of period lengths caused by the segmental effects of obstruents, paired pulsing, vocal fry, the “intrinsic pitch” of sonorants, etc. Those who find this objectionable might want to consider one of the later definitions, but I believe that this one is perfectly legitimate and has many uses.

The second possible definition of production pitch for speech corresponds to a slightly higher level, and is partly motivated by the myoelectric/aerodynamic theory of phonation. This is the standard model of laryngeal sound generation that says that the vibratory cycles come about from the changing trans-glottal air pressure and elasticity of the glottal folds, involving the same Bernoulli effect that occurs in some reed instruments. As a person speaks, he adjusts the tension of the folds to change the frequency (or presence) of these forced oscillations; from a functional point of view this is presumably because the trans-glottal pressure is not susceptible to such exacting control. This tension is determined by several muscles. Although at one time scientists entertained the hypothesis that each closure and opening was a separate, actively controlled neurophysiological event, it is now almost universally accepted that the glottal folds are passive, unwitting participants to the oscillations.

The definitions that this suggests are a description of the laryngeal control signals, or an indication of the vibration rate that would result from those control signals if the transglottal pressure were held constant over many cycles. The micromelody presumably comes about because control of the laryngeal dynamics occurs in ignorance of the actions of the articulators and of the diaphragm. The configuration of the articulators can change the final vibratory pattern because different configurations result in widely varying driving impedances seen by the larynx; similarly, the subglottal pressure source can also change the trans-glottal pressure, though usually not as quickly as the articulators can. This has the very important consequence that unlike the previous definitions of production pitch, the time locations of the glottal cycles are no longer taken to be the crucial determiners of estimates of the pitch: this introduces the notion of a pitch that controls where the cycles “should be,” or “would be,” or “really are.”

---

This characterization has the advantage of potentially being able to describe a
greater variety of phonatory phenomena, and moreover it gives a pitch contour which,
since most of the micromelody is removed, could sometimes be more useful for speech
recognition and linguistic research. That is about all it has going for it, however. First
of all, our current understanding of laryngeal mechanics is too limited for this defi-
nition to be useful. There are actually many muscles and other physiological entities
which have an effect on the state of the folds, and they interact in a complicated way.
Moreover, the concepts of “control signal,” or “state,” or even “muscle tension” are
hard to define, and whatever their definitions, at the moment it is is difficult to ac-
quire any revealing data on them. EMG signals, for example, are too noisy to be very
useful. Instead of figuring out the control signals, one might try to capitalize on the
“What the pitch would have been” aspect, by collecting pitch characterizations (where
pitch” is defined using one of the earlier low-level definitions) from utterances where
a person attempts to use the same intonation with different phonetic content. Such
data are hard to produce and difficult to interpret, and at best they still give only an
uninformative, very incomplete characterization. Secondly, there are problems with
the model that this characterization depends on. As it turns out, the different “mod-
ules” in the speech production process are not actually completely independent; the
tongue root, for example, has a direct physical link to the larynx, and so the actions
of the tongue have an effect on the larynx. Also, it is conceivable that the laryngeal
control adapts to the state of the pressure source beneath it, in order to counteract its
effects. There is also the problem that because of physical noise muscles don’t quite
respond deterministically to the control signals; one way of viewing that fact is that
the model is simply incomplete. Yet another aspect of this characterization that some
people might take to be a disadvantage is that the pitch characterizations one obtains
from it aren’t actually smooth and uneventful. Because the same muscles that put
the folds in a condition that prevents self-sustaining oscillation also play a part when
voicing is going on, one will still see the effects of segments being voiced and unvoiced
in the resulting contours. Even if all these problems could miraculously somehow be
solved, there are still many applications where this characterization is not as useful as
the straightforward glottal closure time characterization.

The third and last of these definitions of production pitch for speech is even more
abstract; there is absolutely no data which applies directly to it. This definition cor-
responds to a representation of whatever it is in the human language faculty that
determines the control signals described in the previous characterization. Some might
call this intonation, or a part of intonation. It should be clear that the best rep-
resentation of this entity is undoubtedly not going to have the form of a frequency
contour, or perhaps even look like a rough schematization of one. There are many
pitch characterizations (by one of the earlier definitions of pitch) which a speaker is
physically capable of producing which in fact he never does produce, with any pos-
sible sentence. The pitch characterizations that he does produce have a systematic
timing relationship with the phonetic segmentation of the utterance. Moreover, differ-
ent pitch characterizations have different meanings and the semantic and pragmatic
context of an utterance will change the possibilities. Any full theory of intonation—again, all characterizations depend on a theory—would have to be able to express these constraints, and of course a simple representation in terms of a time-frequency curve alone cannot do that. Note that unlike the previous characterizations, this one will have to be language specific, since the class of possible pitch characterizations, and the relationship between a pitch characterization and the utterance it is associated with, varies considerably across languages.

Despite the unrelenting attention the subject has received from linguists, the form the characterization should take is still controversial. Though some recent work has solved many important subproblems for at least English intonation [85], the theory is still in its preliminary stages. Even were that not so, my particular interest is with the lower-level characterizations. Being able to obtain this higher-level characterization from a speech signal would be useful in automatic speech recognition, for example, but there are many applications for which it is neither necessary nor sufficient (see Section 1.4.2). It would seem likely anyway that acquiring one of the lower-level characterizations of pitch would be a prerequisite for obtaining this higher-level one.

1.1.1.1 On Voicing

Voicing is subject to the same obscuring influences as is pitch. Fundamentally, this is because in many ways the problems of defining voicing and the problems of defining the pitch of speech signals are the same; for any definition of one there is an analogous definition of the other. This connection is important because it allows the efforts to understand these two concepts to inform and clarify each other. This can be easily seen when one considers the definitions of voicing which correspond to the definitions of pitch so far discussed.

One of the pitch definitions we considered at the lowest level was the glottal closure location representation. The most obvious definition of voicing at this level is that a region is voiced if and only if the glottal closures are spaced close enough together. Unfortunately, this has the consequence that a very reduced vowel with just a single glottal pulse is completely unvoiced, and it also requires the invention of a constant threshold. The alternative to a threshold is saying that the areas between all glottal closures are unvoiced, and that voicing occurs only at the glottal closure instants themselves. Neither of these is very palatable. Instead, it would seem preferable to define voicing physiologically. This means moving up to the next level. Note however, that when we make this move the glottal closures lose their status as infallible indicators of voicing; it is possible that in a transition from a voiced to an unvoiced state there might be some residual flapping of the folds even after the laryngeal muscles have set themselves in “unvoiced” mode, and it is also possible that a very reduced

---

3While it might seem clear from this point of view that there is no good definition of voicing at the “acoustic” level, the statement is not to be taken lightly. It means that the designs of all current v/uv decision makers that I know of are fundamentally misguided.
schwa might appear completely devoiced (no closures at all) even though the laryngeal muscles set themselves into "voiced" mode temporarily.

In both of the previous two cases, the definition of voicing is practically determined by the corresponding definition of pitch.\footnote{The facts that short-term analysis pitch detectors can act independently of a voiced/unvoiced (v/uv) decision maker, and that the structure of the simplest speech production model has a switch between impulses and white noise, might mislead a person into believing that the two issues are independent. They are not; at the lower levels of description they are inextricably linked.} Pitch and voicing only become separated at the next level—possibly called the phonological level. But now we become even more distant from the signal; like all the other discrete units of phonology, phonological voicing becomes smeared at the articulatory level.

There are also potential definitions of voicing corresponding to mathematical pitch and reception pitch: for example, the alternations between periodicity and aperiodicity, or between perceived pitch and no perceived pitch (this statement may make more sense after sections 1.1.2 and 1.1.3). However, although one sometimes hears such definitions, voicing is almost always associated with production.

### 1.1.2 Mathematical Pitch

The next interpretation of pitch to discuss is mathematical pitch. This refers to an a priori notion of periodicity that is independent of the existence of either a source or a receiver of a signal. It is based on some suitable mathematical extension of the idea of the period of a periodic signal, to handle noisy signals whose local period, spectrum, and energy change through time. I believe mathematical pitch to be less relevant to speech than the other two—this will be discussed in Sections 1.2.1, 1.4.1, and 1.4.2. However, a great deal of work on pitch determination is motivated by this interpretation of pitch, and so I want to make it clear both why I believe this motivation to often be unfounded, and why, even when the motivation is correct, many of the implicit definitions of pitch used are not fully satisfactory.

Because my definition of mathematical pitch is rather complicated, I would like to precede it with several simpler, less general, preliminary definitions; hopefully this will help in understanding the final one.

The age-old definitions of periodic and period are concise and elegant:

A function $s$ defined over $\mathbb{R}$ is periodic iff there exists a non-zero $p$ in $\mathbb{R}$ such that $s(t) = s(t + p)$, for all $t$ in $\mathbb{R}$.

$p$ in $\mathbb{R}$ is the period of $s$ iff $p$ is the smallest positive number satisfying this condition. \hfill (1.1)

This is sufficient for investigations in pure mathematics, but it isn’t enough for practical applications in signal analysis. If one were given an infinite signal, its whole domain
would have to be examined to be absolutely certain the true period had been found, or even to determine whether the signal was periodic at all; this clearly can't be done (this sometimes referred to as the "limited aperture" problem). Fortunately, however, for real world signals there is often a reasonable range of possible periods—for speech, we could safely say that .1 msec to 1 second covers any conceivable period—and moreover it is acceptable to consider only a finite set of periods falling in that range—say a million of them. The upper bound on possible periods allows restricting analysis to a subinterval of the signal of width twice that bound, because if no shifts within that subinterval succeed, the signal isn't periodic. The lower bound means that it isn't necessary to consider shifts which are arbitrarily close to zero, so that uniform time quantization does not cause aliasing problems. The fact that the number of periods to try is finite finally makes it possible to attempt the job.

Often, these additions are still not sufficient for dealing with a signal, because many signals are only quasi-periodic, and not periodic.\(^5\) Our present goal is to determine what exactly we mean by "quasi-periodic." One simple possibility is the following:

A stochastic process \(q\) is \emph{quasi-periodic} iff it is the "sum" of a deterministic periodic function \(g\)\(^6\) and a stationary stochastic noise process \(w\).

A deterministic function \(s\) is \emph{quasi-periodic} iff it is a sample function of some quasi-periodic process \(q\).

The \emph{quasi-period} of \(s\) is the period of the corresponding function \(g\) of process \(q\).

This would handle, for example, signals like \(\sin(t) + w(t)\), where \(w\) is a white noise process. Moreover, there are classical techniques which would be of great help in finding the quasi-period of signals. However, this definition does not cover a large class of signals which we would like to call quasi-periodic; a quasi-periodic signal has somewhat periodic behavior locally, but it need not have any such structure globally. Contrary to our intuitions, this definition insists on a global quasi-periodicity, a single quasi-period. Instead, we would like to have a definition which allows a characterization of a signal in terms of the usual auxiliary time-frequency curve, giving the "local fundamental frequency" in the signal. A definition which makes this possible is the following:

A function \(s\) on \(\mathbb{R}\) is \emph{quasi-periodic} iff there exists a periodic function \(g\) with period 1 on \(\mathbb{R}\), and there exists a continuous nondecreasing function \(F : \mathbb{R} \rightarrow \mathbb{R}\) such that \(s(t) = g(F(t))\), for all \(t\).

Then \(f(t) = \frac{d}{dt} F(t)\), when it exists, is the \((local)\) \emph{quasi-fundamental} of \(s\) at \(t\).

When \(1/\frac{d}{dt} F(t)\) exists, it is the \((local)\) \emph{quasi-period} of \(s\) at \(t\).

\(^5\)I have chosen the prefix "quasi-" as being more fitting than "pseudo-.

\(^6\)One might also define this \(g\) to be a random periodic process.
Figure 1.1: One definition of quasi-periodicity gives $2t$ as the quasi-fundamental of \( \sin(2\pi t) \); a second one gives $t$. The first is the correct one, as can be seen by making rough estimates of the apparent local periods.

One way of looking at this is as a generalization of the definition from signal theory of \textit{instantaneous frequency}: the instantaneous frequency of $e^{2\pi \varphi(t)}$ is the derivative of the phase, $d\varphi(t)/dt$. The requirement that $g$ have unit period is necessary to reduce some of the ambiguity of the definition: otherwise $g(t) = g(at)$ and $F_1 = a^{-1}F$ could always replace $g$ and $F$, for all positive $a$. It is quite tempting to do this definition another way: if $g(t)$ is a periodic function with period 1, then $f(t)$ gives the quasi-fundamental of $s(t) = g(f(t)t)$. For periodic $s$ (constant $f$) this amounts to the same thing, since $\frac{d}{dt} f(t) = f(t) + t \frac{df}{dt} = f$. Otherwise they are different, and it is clear that the first definition is the correct one (see Figure 1.1).

There are cases where a signal $s$ can have more than one pitch contour, by this definition. Consider, for example, the $s$ resulting from any $g$ and $F$ satisfying the conditions above, plus the condition that $\frac{d}{dt} F(t)$ be periodic with period $p_1$, such that $p/p_1$ is rational. Then $s$ is not only quasi-periodic, it is also periodic, so that not only does it have the pitch contour from $g$ and $F$, it also has the one from $g_1 = s$ and $F_1(t) = t$. An example of this is $g(t) = \cos(w_1 t)$, $F(t) = w_2 t + \sin(w_2 t)$, $g_1(t) = \cos(w_1(w_2 t + \sin(w_2 t)))$, and $F_1(t) = t$, with $w_1/w_2$ rational. ($g$ and $F$, and $g_1$ and $F_1$, have to undergo the transformation to make $g$ and $g_1$ have a unit period, but the two pairs will still be different.) These examples are not so extreme either; such situations could quite conceivably arise from trills, vibratos, or tremolos. The definition is simply reflecting a genuine ambiguity in what we mean by “local periodicity.” The definition doesn’t give much of a handle on how to limit or examine the possibilities, but one might imagine using such rules as “given a choice, choose the $g(t)$ with the shorter (longer) period” or “given a choice, choose the $dF/dt$ that is (is not) periodic.”

This definition does capture the notion of “pitch contour,” and I will have use for it later on when I consider the problem of converting glottal closure labels to pitch contours (section 1.2.3), but it still leaves something to be desired for mathematical pitch. It has no provision for noise, nonstationarity, etc.; doing something like combin-

\footnote{Note however, that this ambiguity has no effect on the “pitch contour” $\frac{d}{dt} F(t)$, because the transformations cancel each other. If $g$ has period $p$, the quasi-fundamental is $(1/p) \frac{d}{dt} F(t)$.}
ing this definition with the previous one actually accomplishes little towards this end. This definition says that all quasi-periodic signals can be obtained by time warping (stretching and contracting the time axis) of a periodic signal; it lets the period vary, but it doesn’t allow enough variation in short-term energy or spectrum.

Since the early 19th century, we have known that a periodic signal can be represented as a sum of sinusoids whose frequencies are multiples of the signal’s fundamental frequency (the reciprocal of its period). To determine the harmonics in a periodic signal, we can take its Fourier transform; the frequencies at which the transform is non-zero are the harmonics, and the values of the transform at the harmonics give their amplitudes and phases. Thus from an abstract point of view it is quite easy to find the unknown fundamental of a given periodic signal: first determine what the frequencies of its harmonics are; the fundamental is then the greatest common divisor (GCD) of those frequencies. Note that GCD is exactly the right operator; “common divisor” is equivalent to the $s(t) = s(t + p)$ condition above, and “greatest” is equivalent to the condition that $p$ be the smallest satisfying the first condition.

Generalizing this insight to give a definition of quasi-periodic is not difficult:

\[
\sum_{k=1}^{\infty} A_k(t)e^{2\pi k F(t)} ,
\]

where (i) $F(t)$ is continuous and nondecreasing, (ii) the $A_k(t)$ vary “more slowly” than $F(t)$, and (iii) for all $t$ the $k$ for which $A_k(t)$ is non-zero have no common divisor other than 1.

In that case $f(t) = \frac{d}{dt} F(t)$ gives the quasi-fundamental of $s(t)$ at time $t$.

The frequencies $k f(t)$ for which $A_k(t)$ is non-zero at time $t$ are called the instantaneous harmonics of $s(t)$ at time $t$.

The condition on the variability of the $A_k(t)$ is an attempt to reduce the ambiguity in the definition; clearly any $s(t)$ can be represented by just $A_1(t)$ and $F(t)$ if $A_1(t)$ isn’t constrained. The condition on there not being a non-trivial common divisor of the instantaneous harmonics is to make certain that $f(t)$, and not a multiple of $f(t)$, is the quasi-fundamental. This definition could be viewed as a reformulation of the previous one to allow for time-varying spectrum and energy. This allows harmonics to disappear and reappear, to change in magnitude, and to move around. It still does not allow for noisy or distorted signals, but I believe that the best way to deal with those phenomena is to say that such signals are not quasi-periodic, but that they are made up of a quasi-periodic component and a non-quasi-periodic component. Finding the quasi-fundamental of those signals amounts to finding the quasi-fundamental of their quasi-periodic component. Signals with inharmonic components are not quasi-periodic, but I don’t think they should be; steady state versions of those signals certainly aren’t called periodic. Note that like the previous definition, this one is ambiguous; if, for example, $s(t)$ is a phase-modulated square wave, then $f(t)$ could be either a raised square wave, oscillating between two fundamments periodically, or it could be a constant equal to the oscillation rate.

This definition (like any other) could be considered to entail two claims: it deter-
mines the class of quasi-periodic signals, because of course not all signals should be
called quasi-periodic, and for the quasi-periodic signals, it describes what their quasi-
period is through time. In criticism of the definition, it might easily be claimed that
it is too general, that by playing around with the $A_k(t)$ and with $F(t)$ it is possible
to construct signals $s(t)$ which shouldn't be quasi-periodic or which shouldn't have
the quasi-fundamental $f(t)$. The condition on how quickly the $A_k$ may vary attempts
to solve this problem, but obviously the requirement (perhaps "guideline" is a bet-
ter word) is quite vague. So far, however, I have been unable to discover a criterion
providing a hard cutoff; there appears to a continuum of gradually less reasonable
cases.

There are many issues that must be dealt with in an implementation of this def-
inition. If it is implemented using one of the usual short-term spectrum estimation
methods, then there will be the usual problems associated with a window; the use
of a window entails an assumption about the possible range of values for the quasi-
fundamental, and the window will cause the peaks (underlyingly, impulses) in the
spectrum to be smoothed and distorted. The computation probably will implicitly
assume that the signal is periodic within the window, and since the signal is only
quasi-periodic, some averaging will occur and the harmonics will be shifted, probably
enough to make them inharmonic. Because of quantization, noise, and windowing, the
peak locations will not be found exactly; there might even be a few spurious peaks
in the spectrum which are indistinguishable in their local properties from others. A
quasi-GCD algorithm will have to be found which will work in these circumstances.
These questions will be addressed in section 2.2.

1.1.2.1 Alternative Definitions

As I mentioned at the beginning of this section, a great deal of the past work
on pitch determination has been underlyingly motivated by mathematical pitch: a
conception of pitch which has something to do with local periodicity in the signal.
The algorithms which are produced by this approach all work by doing a short-term
analysis of successive regions of the signal, computing an approximate average value of
the period within each of the regions. Just as nonstationarity can sometimes be dealt
with by excising out short portions of the signal and pretending the signal is stationary,
so too one can attempt to deal with quasi-periodicity by excising and assuming that the
signal is (close to) periodic. Members of this class of pitch determination algorithms
are naturally labeled by Hess ([46]) as "short-term analysis" algorithms. Because these
algorithms are designed with mathematical pitch in mind, one might wonder whether
the methods in some of them could be used as the basis of a general definition of
quasi-periodicity. The purpose of this section is to demonstrate that this is not the
case; i.e., that unlike the other two classes of pitch definitions, so far it looks like

---

8I use the phrases "mathematical pitch" and "quasi-periodic" almost interchangeably, since I know of
no other concepts besides quasi-periodicity which might fall under mathematical pitch.
mathematical pitch corresponds to only one definition: one similar to, or equivalent to, definition 1.4.

What follows, then, is an examination and criticism of a representative sample of the past pitch determination work which can be subsumed under the heading of mathematical pitch. Although in some ways it would be preferable to examine these algorithms in the next chapter, in which I am more directly concerned with implementation, a lot of the difficulty in constructing algorithms for mathematical pitch is not in the implementation so much as it is in trying to determine what to implement. This is just the opposite of production pitch; the definitions given in section 1.1.1 were readily apparent, but implementation of them is quite difficult. Defining quasi-periodicity is not easy; by examining the alternative definitions corresponding to these algorithms, I hope to provide further motivation for the definition that was given above, and to reveal the consequences of choosing something different from it. I would like to emphasize that my interest throughout this section is in the definitions of pitch underlying the algorithms, not in the implementation details. The criticism of the algorithms that is given here is based primarily on the definitions of quasi-periodicity which they suggest; the shortcomings which they exhibit as definitions may not necessarily have any important effect on their performance in the uses to which they are normally put. They might be quite useful in practice; in fact, they could conceivably even be superior to direct implementations of better definitions, for a restricted class of input signals. However, this criticism can often provide a principled explanation for the failings which do show up, and an awareness of a good definition of quasi-periodicity can provide guidance in improving algorithms.

I should make an important point which applies to all my discussions of other people’s pitch determination algorithms (pitch trackers). Not only do most of the papers in the literature not explicitly state what they are trying to achieve, they often give contradictory clues; it is thus entirely possible that some authors of pitch trackers would not agree with my analysis of what definition of pitch their work is assuming.

One of the simplest and oldest pitch trackers is one based on short-term autocorrelation or cross-correlation ([109], and many more). This is a member of a general class of trackers which presuppose the following definition of mathematical pitch:

Given $s(t)$, perform the transformation $T(s, t)$ to get the function $S(t, p)$.

The local quasi-period of $s$ at $t_0$ is the number $\hat{p}$ that maximizes $S(t_0, p)$ \hspace{1cm} (1.5)

over values of $p$ in its domain.

Typically $T(s, t_0)$ considers the values of $s$ only in a neighborhood around $t_0$, and $S(t_0, p)$ is only defined for $p$ in a finite interval. Implementations of the transformation $T$ might for example be based on the Fourier transform, on the cepstrum, on autocorrelation, or on the average magnitude difference function (AMDF, described below).

Correlation is simple, intuitive, and widely used; moreover, it turns out that the problems with using it as part of a definition of quasi-periodicity crop up in different guises in most of the other algorithms as well. For these reasons I will spend a bit more time on correlation than on the subsequent algorithms. In correlation, $S(t_0, p)$,
Figure 1.2: The upper plot is of a short portion of a steady state /c/, from a female speaker, with a quasi-fundamental of roughly 235 Hz. The lower plot is a plot of the normalized sequence $c(n)$, described in the text. The value of $L$ in the calculation was 20 msec. The correlation sequence can also be computed by taking the inverse DFT of the squared magnitude of the DFT of the subsequence of $x(n)$; the result is similar to the one plotted here.

for each $p$, is found by taking the dot product of a fixed windowed portion of the signal around $t_0$ with another portion of the signal $p$ seconds away. This is appealing because it would seem that it couldn’t possibly go wrong: one might think that the shift which gives the highest correlation has to be the right answer, and that the subpeak heights in $S(t_0, p)$ must reflect how good the other shifts $p$ are. As we shall see, however, these thoughts are not altogether justified.

There are several ways that correlation can be implemented (see [89]); one straightforward method is to calculate the sequence $c(n) = \sum_{i=n_0}^{L+n_0} x(i) x(i+n)$, for $N_l < n < N_u$.

Here $n_0$ is the index in the sequence $x(n)$ where an estimate of the local period is desired; $N_l$ and $N_u$ are the lower and upper bounds on possible periods; and $L$ is the length of the subsequences used. The period is presumably the index $\hat{n}$ which maximizes $c(n)$. Often $c(n)$ is normalized by the norms of the two vectors of length $L$, $x(i)$ and $x(i+n)$, which are involved in the dot-product; $c(n)$ is then the cosine of the angle between the vectors. An example of this computation is show in Figure 1.2. Even though for this example the biggest peak in $c(n)$ in the range (.003, .02)—a reasonable range for speech—is (approximately) at the correct answer, it is clear that the $c(n)$ sequence is not at all as unambiguous as initial intuitions might have lead one to
believe. The reasons for the problem are not hard to find. It is easy to show that:

\[ g(t) = \sum_{n=1}^{\infty} a_n e^{2\pi n ft} \implies S(p) = k_1 \int_0^{1/f} g^* (t) g(t + p) dt = k_2 \sum_{n=1}^{\infty} |a_n|^2 e^{2\pi n f/p}, \]

where \( S(p) \) is \( g \)'s long-term autocorrelation function.\(^9\) Thus for a periodic input with period \( p \), \( S \) will have absolute maxima at 0, \( p, 2p, 3p, \ldots \). In fact, \( S \) itself will be periodic; \( S(p) \) is basically just a zero phase version of \( g(t) \), with the amplitudes of the harmonics squared. Thus \( S(p) \) does not have a single clear peak at the answer, even for cases which one would think should be unambiguous. The source of this problem is a failure to model both of the two requirements for being a period, that were given in definition 1.1: i) \( s(t) = s(t + p) \), and ii) \( p \) is the smallest positive number satisfying (i). Obviously, if \( \hat{p} \) is the period, then not only is it true that \( s(t) = s(t + \hat{p}) \), but also \( s(t) = s(t + n\hat{p}) \), for all integral \( n \), so that this \( S \) accurately reflects only how well different shifts \( p \) satisfy requirement (i). The peaks in \( S(p) \) at \( n\hat{p} \) are entirely due to the fact that \( \hat{p} \) is the period, and they would not be there if \( S(p) \) properly reflected the fact that a period is the smallest number satisfying \( s(t) = s(t + p) \). When requirement (ii) is added there should never be any peak at \( p = 0 \), and when the correlation at a shift \( np \) is the same as that at shift \( p \), there should be no peak in \( S \) at \( np \). Finding \( p \)'s for which \( s(t) \) and \( s(t + p) \) are roughly equal is easy; in fact is hard to see how one could improve upon something like correlation. We don't need to improve on condition (i); the mistake is in stopping with just that condition, without considering how the "smallest \( p \)" criterion should be incorporated.

Another way of seeing the problems with using correlation in a definition of quasi-periodicity is by considering the following requirement on \( S \), for definitions of mathematical pitch which follow the pattern of definition 1.5: the function \( S(t_0, p) \) should reflect our intuitions about mathematical pitch, by taking on values which are monotonically related to the likelihood that the points \( p \) are the local quasi-period for \( t_0 \).\(^10\) This requirement seems reasonable, in light of the independence assumption inherent in the "peak-picking" definition of quasi-period, and it certainly would seem to make any implementation of this definition more robust than if this were not so. I should clarify that what I am referring to here is the likelihood that each \( p \) is individually the sole quasi-period of the input; to represent the possibility of two or more quasi-periodic functions being mixed together would require a higher dimensional \( S \). Note that if we had a dependable way of getting \( S \)'s which satisfied this requirement, for applications requiring only a visual display it wouldn't even be necessary to determine the pitch

---

\(^9\)This of course is why the sample autocovariance of a signal can be used in estimating its power spectrum.

\(^10\)Note that though this radically restricts the possibilities for \( S \), it does nothing to restrict the possible effective definitions of quasi-periodicity. The precision and exhaustiveness of our intuitions determines how many possibilities for \( S \) this requirement leaves for a given \( f \); we must be careful that our intuitions are not inconsistent.
contour from \( S(t_0, p) \) (by finding the maxima for each \( t_0 \)); we could just display \( S(t, p) \) like a spectrogram. When the \( S \) that correlation determines is displayed like a spectrogram, the inadequacies of correlation become evident. Speech is often "strongly quasi-periodic," and for such signals the display should have a clear ridge following the quasi-period, and there shouldn’t be other peaks and ridges at anywhere near the same height. But for short-term autocorrelation, this display will have several parallel ridges, each following a different multiple of the quasi-period, and it will have many subsidiary peaks corresponding to the formants (witness the big effect of the harmonic amplitudes on \( S(p) \) above).

As with all the algorithms of this section, the signal processing requires that an assumption be made that the sequence within the analysis region be close to periodic. Note, however, that this method would fall apart on a truly periodic input; it crucially depends on a slight aperiodicity to make the peaks in \( c(n) \) at higher multiples of \( p \) slightly lower—this can be seen in Figure 1.2. The instability of the periodic case can easily fall on the other side as well: if it ever occurs that two periods are together even slightly more similar to the next two periods, than the first period is to the second, this method will incorrectly label the quasi-period to be twice what it should be.\(^1\) Because the peaks in \( c(n) \) seem so much at odds with what one would expect, it is hard to see how correlation could be made part of a reasonable definition of quasi-periodicity. For certain quasi-periodic signals, the spurious peaks in \( c(n) \) become higher than the intuitively correct one, which would be intolerable if correlation were considered to be the definition of the quasi-period.

These problems with spurious peaks do actually show up when the straightforward version of correlation described above is implemented and used on speech signals. This has motivated people to modify the method to make it work more dependably. These modifications are of four types. The first is to use fancier methods of determining the peak to select in \( c(n) \). Instead of simply choosing the highest peak in a fixed range of shifts, most implementations choose the first "big" peak, or the first peak above a threshold; moreover the values of \( c(n) \) at \( p/2, p/3 \), etc. are checked before accepting period \( p \). Secondly, the sequence \( c(n) \) may be changed: short-time autocorrelation rather than cross-correlation might be used ([28], [89]); the sequence \( c(n) \) might be weighted to penalize greater shifts \( n \); or the sequence \( k(n) = \sum_{i=1}^{M} c(in) \) might be used instead of \( c(n) \). Thirdly, pre-processing of the input signal \( x(n) \) might be used to improve the appearance of \( c(n) \) ([109], [89], [57], among others). In particular, whitening by center clipping or inverse filtering is used to get rid of some of the harmonic amplitude effects in \( S(p) \). (Note that this does nothing to solve the period multiple problem—in fact it might make it worse, by making the signal less time-varying.) Fourthly, post-processing might be used on the sequence of period estimates ([99], [100], [92], [69], [110], [88], among others). Estimates that are obviously inconsistent

\(^1\)This error is related to, but is not necessarily caused by, the so-called "pitch-doubling" that sometimes occurs in speech production. There is an ambiguity here, but this method chooses the wrong point to switch between the two cases.
with the values from neighboring analysis regions can be replaced.

A combination of these changes will produce a pitch tracker which works fairly reliably on speech signals—a restricted subset of the quasi-periodic signals. However, the reasons for rejecting correlation as part of a general definition of quasi-periodicity still remain; the modifications only gloss over the fundamental definitional problems described above. Fancy peak-picking methods should not be necessary in the $S(t_0, p)$ that we want; when it is difficult to find peaks, it should be because there is a genuine ambiguity in the signal, thus perhaps calling into question the assumption of quasi-periodicity in the first place. As for changing $c(n)$, it is not hard to work out signals which will elude the attempts listed; even if the counterexample signals do not occur in speech, their existence still means that the definition is not fully general. The pre-processing techniques make special assumptions about the signal; this further vitiates the status of correlation in a general definition of mathematical pitch. Complicated methods of using context constraints to correct the algorithm shouldn't be necessary. Correlation fails on signals which should be unambiguous and simple to process; with a pitch tracker based on a better notion of mathematical pitch, one might expect that a simple-minded post-processing would work if any were necessary at all.

In retrospect, it would have been quite surprising if it had been possible to completely resolve the difficulties with correlation, because the correlation sequence $c(n)$ isn't all that different from the original sequence $x(n)$: it too is quasi-periodic, and in almost the same way. For many signals, finding the correct peak in the correlation sequence is not that much simpler than finding the period directly in the original unprocessed signal.

Having spent some time discussing correlation, we can move more quickly through the other techniques.

Clearly the relatives of correlation, which simply use a different operator instead of multiplication in the dot product, manifest the same broad level characteristics as correlation. The most prominent example of this is AMDF (Average Magnitude Difference Function, [66], [95], [122]), which uses magnitude difference instead of multiply. Correlation and AMDF differ in computation time, robustness in noise, and accuracy. These fine details are important for an implementation, but they hardly matter for the question which concerns us now. Other slight modifications of correlation, such as using dynamic time warping ([70]), also do not affect the definitional issues much.

The so-called "maximum likelihood" approaches ([75], [31]), so far at least, end up with a situation extremely similar to correlation; in fact, the equations are almost identical. As suggested by definition 1.2 above, in these methods a signal $s(t)$ is locally modeled as $g(t) + w(t)$, where $g(t)$ is an unknown periodic function, and $w(t)$ is a known noise process. The quasi-period for that portion of $s(t)$ is then the period of the $g(t)$ that is most likely to have produced $s(t)$ in the noise. Alternatively, formulated without appealing to probability, the goal is to find the least cost approximation (by some chosen distance criterion) to the signal by a member of the periodic signals, and the local quasi-period is the period of that member. The same problem turns up here as for correlation, in that the effective definition of "period" for $g(t)$ only takes
into account the first requirement, so that if there is a function \( g_1(t) \) with period \( p \) that fits \( s(t) \) well, there will also be a \( g_2(t) \) with period \( 2p \) that will work as well, and probably better. This is exactly what is seen when these are implemented; the peaks get gradually higher for bigger and bigger multiples of \( p \), unless something is deliberately added to the equations to offset the affect (see [46] p.446-466, for a good overview of this approach).

Some of the oldest approaches to pitch tracking have been based on the Fourier transform. The simplest proposals would be that the quasi-fundamental of a signal is at the frequency of the biggest or the first peak in the short-term (but narrow-band) spectral slice of a signal. It should be clear that this is no good as a concept of quasi-periodicity: the function \( \sum_{j=2}^{5} \sin(2\pi 100 j t) \) confounds both of them. Methods which attempt to accentuate the fundamental by pre-processing the signal with a (fixed) nonlinear function ([46] p.302-342, [63]) are only successful for a fairly restricted class of signals; as with correlation, the efforts to improve the basic method do not effect its possibilities as a definition of quasi-periodicity.

It seems clear that all of the information about the harmonics will have to be considered and combined if the Fourier transform is to be used to find the quasi-fundamental. One method that attempts to do this is one based on the cepstrum: the quasi-period is the location of the biggest peak in the inverse Fourier transform of the log of the short-term magnitude spectrum; that is, the quasi-fundamental is the frequency of the biggest sinusoidal component in the log spectrum ([73], [74]). Although there is some motivation for this definition of the fundamental of speech signals, it is certainly a strange general characterization of the spectra of periodic signals, and so as one would expect, it does strange things with periodic signals which don’t fit that characterization. The cepstrum depends on there being many harmonics in the spectrum; the cepstrum of a pure sinusoid is another sinusoid—hardly an unambiguous representation of the period. It also depends on there not being any regular pattern in the magnitudes of successive harmonics, to keep the peaks in the cepstrum at the subharmonics of the fundamental frequency from being too high. A signal having only odd harmonics\(^\text{12}\) will have its biggest magnitude peak in the cepstrum at half of the true period. All of these examples should be easy to handle by a transformation which corresponds to a useful notion of quasi-periodicity. For speech signals, it is often possible to get by with the assumptions inherent in the cepstrum, and in fact cepstral pitch trackers were for a while regarded as the standard. However, we have seen that the cepstrum is out of the question as the basis for a definition of mathematical pitch.\(^\text{13}\)

\(^{12}\)There is some inconsistency in how harmonics are numbered in the literature. Some people say that the first harmonic of a tone is twice the fundamental; I have heard some people state that this is where the term “f0” came from. Such a numbering scheme has the consequence that the odd harmonics are the even multiples of the fundamental, and the even harmonics are the odd multiples, and so I call the fundamental the first harmonic. This is almost standard.

\(^{13}\)It is interesting to compare the cepstrum to the signals obtained if functions other than log are applied
There are many other ways of combining the harmonics in the spectrum. Some transforms that naturally fall together are the harmonic product spectrum, the log harmonic product spectrum, the harmonic sum spectrum, the frequency histogram, and period histogram ([75], [98]). The first three of these are given by:

harmonic product spectrum:

\[ P(e^{j2\pi f}) = \prod_{n=1}^{N} |F(e^{j2\pi n f})|^2 \]

log harmonic product spectrum:

\[ L(e^{j2\pi f}) = \log(P(e^{j2\pi f})) = 2 \sum_{n=1}^{N} \log(|F(e^{j2\pi n f})|) \]

harmonic sum spectrum:

\[ Q(e^{j2\pi f}) = \sum_{n=1}^{N} |F(e^{j2\pi n f})|^2 \]

where \( F(e^{j2\pi f}) \) is the Fourier transform of the signal, and \( N \) is an arbitrary number (5 is used in Noll ([75]) and in Schroeder ([98])). The frequency histogram is the same as \( L \), except that it quantizes the Fourier transform to be 1 at peaks and 0 everywhere else. The period histogram is equivalent to the frequency histogram. Obviously \( P(e^{j2\pi f}) \) has the same properties as \( L(e^{j2\pi f}) \). Thus I will just discuss \( P \), the harmonic product spectrum, and \( Q \), the harmonic sum spectrum. For periodic signals whose first \( N \) harmonics have significant amplitudes (which is sometimes approximately true of speech), the product spectrum will have a single sharp peak right where it should be (see Noll). But the price that is paid for this wonderful signal to noise ratio is that the transform falls apart when the assumption that those harmonics are all there, is violated. If a single one of those harmonics is zero or close to it, say the \( r \)'th one, then the \( |F(e^{j2\pi r f_0})| \) term in the product will cancel the peak at \( f = f_0 \) in \( P(e^{j2\pi f}) \). The harmonic sum spectrum is more forgiving of this transgression, but it pays the price for this laxness by putting in peaks where none should be. For example, if the input were \( \sin(2\pi 40t) + \sin(2\pi 60t) \), \( Q \) would have large amplitude peaks at 10 Hz and at 30 Hz which wouldn’t be present in a more perspicuous representation of the local fundamental. Both of the transforms have an unavoidable trade-off involving the \( N \) parameter. No matter how large \( N \) is chosen to be, neither of these transforms will correctly find the quasi-period of signals whose harmonics don’t have significant amplitudes until after the \( N \)th one: \( \sin(2\pi N f_0 t) + \sin(2\pi (N+1) f_0 t) \) is an example. But if \( N \) is made too big, then the product spectrum will be zeroed out (since not all signals will have harmonics up to \( N \)), and the sum spectrum will have peaks everywhere and will be uninterpretable (for example, the peaks at \( f_0/2 \) will all be of equal height for small \( i \) if \( f_0 \) is the quasi-fundamental). Thus, as with the preceding algorithms, these do not suggest very good definitions of quasi-periodicity, even if for some signals they perform adequately.

to the spectrum. Let \( g(t) \) be a signal, \( G(e^{j\omega}) \) be its Fourier transform. \( \mathcal{F}^{-1} \) be the inverse Fourier transform operator. Then \( \mathcal{F}^{-1}G(e^{j\omega}) \) gives \( g(t) \) back, \( \mathcal{F}^{-1}|G(e^{j\omega})| \) gives a zero-phase version of \( g(t) \), \( \mathcal{F}^{-1}|G(e^{j\omega})|^2 \) gives the autocorrelation of \( g(t) \) (in fact this is sometimes how it is computed), and \( \mathcal{F}^{-1}\log|G(e^{j\omega})| \) (or \( \mathcal{F}^{-1}\log|G(e^{j\omega})|^2 \)) gives the cepstrum.
Yet more transformations based on the Fourier transform use the idea of a "comb filter" ([66], [59], [60], [61], [78]). Note that one of the easiest methods of doing this is actually nothing new.\footnote{[70] and [66] also both note the connections between some superficially disparate pitch determination algorithms.} One way of looking at correlation is that a different comb filter is associated with each of the possible quasi-periods; the value of $S(t_0, p)$ is the short-term energy in the output of the $p$th filter. These different filters correspond to the difference equations $y(t) = x(t) - x(t-p)$; their impulse response is $\delta(0) - \delta(-p)$ and the magnitude of their system function is $|1 - e^{j2\pi pf}| = 2(1 - \cos(2\pi pf))$. In the equation $\sum(x_1(i) - x_2(i))^2 = \sum x_1^2(i) + \sum x_2^2(i) - 2 \sum x_1(i)x_2(i)$, for $x_2(i) = x_1(i-p)$, the LHS is the signal currently being described, and the last term of the RHS is the correlation. Depending on how the short-term autocorrelation is performed, the first two terms of the RHS are constant or close to constant for all $p$. Thus the only difference between correlation and the current method is the sign of $S(t_0, p)$, so that minimizing must be done instead of maximizing. Rather than compute the short-term autocorrelation at a lag $p$ in time, by convolving the signal with the impulse response of $p$'s comb filter, the energy of the result could be found by multiplying the signal's short-term Fourier transform with the system function of the comb-filter, and then finding the energy of that instead (Parseval's theorem). Thus this approach has exactly the problems I described for correlation, and provides another explanation of some of them. The expression $(1 - \cos(2\pi p/n))$ clearly has all of its zeroes at $f = k/p_0$, for $k$ an integer. Thus this filter would entirely cancel out a periodic signal with period $p_0$, and $S$ would have its lowest possible value at $p_0$. However, the comb filter corresponding to $p' = p_0/n$, for $n$ any integer, will also have zeroes at those locations, and so $S$ will have an equally extreme value at all those $p'$. Though this attempt doesn't work, there are other transformations of the input signal that one might consider which use this idea of multiplying a short-term spectrum with different harmonic templates. One idea would be to attempt to emphasize the harmonics rather than cancel them, by using "harmonic sieves" as the templates ([59], [60], [61], [29], [107], [106]). One could, for example, have the template for quasi-fundamental $f_0$ have equal amplitude impulses at the frequencies $nf_0$, and be zero elsewhere. The quasi-fundamental for a short-term spectrum (spectral slice) is the one whose sieve gives the biggest inner product with the slice. Note that this suggestion is identical to the harmonic sum spectrum for $N$ very large. It has the same problem that virtually all the preceding potential definitions have: for a periodic input with fundamental $f_0$, this transform has equal amplitude peaks at all $f_0/n$ (in the lag domain, there are peaks at $np_0$). Martin ([59], [60], [61]) chooses to solve this difficulty by making the amplitudes of the impulses in the sieve decrease with frequency. But such a solution has to balance the tradeoff between too fast a decrease, which will ruin its response to signals with only high harmonics, and too slow a decrease, which won't solve the subharmonic problem. Consider, for example, the signal $a\sin(2\pi 2f_0t) + b\sin(2\pi 3f_0t)$. Suppose that the amplitudes of the first three teeth are $t_1$, $t_2$, and
Then all we need to make the system erroneously respond with \(2f_0\) is to have 
\[at_1 > at_2 + bt_3.\] If \(t_i\) is a decreasing sequence, this is always possible. If they do not decrease, the system will confuse the fundamental \(f_0\) with \(f_0/2\). The importance of this example is not so much its particular interest, but that it exemplifies the principle whereby counterexamples can always be constructed. This is just a slightly less extreme form of the \(N\) trade-off for the harmonic sum spectrum.

Martin’s algorithm works quite well on speech signals; however, the problems it has with even the simple periodic signals above cause it to be unusable for quasi-periodicity. As we have similarly observed for the previous algorithms, modifications of Martin’s idea do not change it basic characteristics; for example, Paliwal and Rao’s addition of amplitude normalization (by low-order LPC spectrum) to Martin’s technique does not remove the \(f_0/n\) problem. Martin’s sieves all have the same amplitude sequence for the teeth; the only parameter that varies is the spacing between them. Another way a person might try to fix the \(f_0/n\) problem is to have equal amplitude teeth, but to change that single amplitude for different combs. One principle for setting the amplitude could be to make all combs have equal energy, because the dot product makes the algorithm biased against combs with lower energy. Another approach might be to remove some of the teeth which correspond to higher harmonics. Neither of these attempts does much better at solving the basic problem than Martin’s does.

The next logical step after combs and sieves is full-fledged pattern recognition. This has not to my knowledge been reported on in the pitch determination literature, though the approach has been used for voiced/unvoiced decision makers ((10), (104), (105)) and of course in speech recognition. However, I would like to bring it up in order to show its pitfalls. In essence, this method is nothing more than a recording of the data. Virtually anything that has been discussed so far could be chosen as the form of the input: cepstra, correlation sequences, spectra, even original signal portions. For a large number of these inputs, the correct quasi-fundamental is stored. The process of pitch detection is then nothing more than finding the entries in the database closest to the new inputs, and returning the quasi-fundamentals stored at those entries. In this view, pitch detection is the problem of approximating a function from a vast set of vectors, into \(\mathbb{R}\). The way the approximation is done is by memorizing enough of the mapping that the value of the function for any vector is pretty close (according to some external criterion) to the value of the function at the memorized vector closest to the given vector (according to some simple distance function).

This solution is devoid of interest from the point of view of someone trying to define quasi-periodicity. It doesn’t represent a definition—or if it does, it is an extremely trivial one—it is only a reproduction of the data. The job of a theory is to find a system, an organization in the distribution which here is just being memorized.

Although I said earlier that I would only discuss algorithms with regard to their definitional possibilities, because this approach has not been (publicly) tried, I feel it would be useful to make a short digression and examine its potential in a pitch tracker. There are many features which make it seem attractive at first. For one thing, this approach seems to be closer to how humans determine the quasi-fundamental from,
say, a correlation sequence. We don’t simply look for a significant peak; instead of the peaks at the multiples of the answer being competitive and damaging, we use them to help us find the correct peak. If in fact we didn’t see those “competing” peaks, we would be very surprised; instead of making an incorrect independence assumption in peak-picking, we take advantage of the dependence in the signal. Another reason the approach looks good is that because we are using the whole vector, the correlation sequence (or whatever) no longer has to be sampled as finely as it did when we depended on the resolution of a single peak for the resolution in our estimate. Now the two issues are separate; we can have a resolution in our estimates of the quasi-fundamental or quasi-period which is much finer than the quantization of the input vectors. A third reason is that this fixes a problem with sieves and combs. The idealized impulse spectra used as the templates would never actually appear as input vectors; a “matched filter” isn’t being used. By using actual input vectors as patterns, we avoid such problems as windowing effects on spectra, etc. Finally, it would seem that the algorithm just couldn’t possibly make a mistake.

The fly in the pudding, I think, is that too many patterns have to be stored to make it work; one can’t get along without a theory in this case. We saw how for Martin a single pattern for each fundamental is not enough; the way to solve the $f_0$ vs. $2f_0$ ambiguity of the two harmonic counterexample given above is to have another additional pattern for $f_0$: one that matches the particular input so well that the differences between the previous two patterns is irrelevant noise. Not only do there have to be many patterns with different combinations of harmonics; something has to be done about the possibilities for variation in the relative amplitudes between the harmonics as well. Another way of looking at this need is that the operation of finding the fundamental is nonlinear; the response to $\sin(2\pi 2f_0 t) + \sin(2\pi 3f_0 t)$ has little to do with the sum of the response to $\sin(2\pi 2f_0 t)$ and to $\sin(2\pi 3f_0 t)$. Thus many patterns have to be added, to avoid the consequences of the fact that the fundamental operation of most pattern recognition algorithms is the dot product—a linear function. Having all these patterns quickly leads to terrible inefficiency in both space and time; the patterns have to be stored, and every input has to be compared to many of them. Some savings could be obtained by restricting the pattern repertoire to those that occur in speech, by normalizing the input to reduce the variability, by using continuity constraints to narrow the range of comparisons, by taking advantage of the distance between stored patterns to eliminate many from consideration at once, and perhaps by using sequential decision theory. But it would seem unlikely that an accurate or fast algorithm could be produced; this approach is clearly inferior to one which is more “data-directed.” Things would be fine if, for example, the spectra of periodic signals formed a subspace of all spectra, with orthonormal basis vectors corresponding to each of the possible fundamentals. In that case, these comparisons wouldn’t be necessary; we could just take the projection of the input onto the basis set and find the biggest component. Note that in essence, this is what cepstrum and sieve methods are trying to do: in the cepstrum, sinusoidal spectra are taken as the orthogonal basis; in a sieve, impulse train spectra are taken as the orthogonal basis.
Continuing onwards, instead of trying to combine the harmonics of a signal by some sort of blind computation, as the methods we have so far seen have tried to do, a different route would be to first figure out where the harmonics are (this wasn’t necessary before), and then to determine an estimate of the fundamental from knowledge of the harmonics. I will leave the question of how the harmonics are detected to section 2.2, because that is entirely an implementation issue; now I would just like to examine how the harmonics should be treated once they are found. The simplest idea that has been tried is to take as the fundamental the most commonly occurring difference between successive harmonics ([8], [101]). This idea of quasi-periodicity is little better than the idea of simply taking the first harmonic; on a signal with only odd harmonics, the correct answer doesn’t even appear among the differences.

We have now gone almost full circle, because when one considers more complicated ways of combining the harmonics, one is soon lead to algorithms based on the definition of quasi-periodicity I gave earlier: the quasi-fundamental is the quasi-GCD of the locations of the harmonics. None of the other methods we have looked at seems to be even close to corresponding to an acceptable definition of quasi-periodicity. To my knowledge there has been only one explicit instance of a pitch determination algorithm based on this idea ([113]), although there have been several instances of similar ideas ([4], [98], [39], [40], [36], [79], [117], [119]). These will be examined in more detail in section 2.2, in which I present one way of implementing this definition.

1.1.3 Reception Pitch

The last of the three classes of pitch definitions is reception pitch. In some ways, this is—both physically and philosophically—the mirror image of production pitch. In both cases, the “pitch” of a signal is a characterization, corresponding to that signal, of the internal processes of some mechanism. The difference is in the question of causality: in one case this correspondence originates from the fact that the mechanism is producing the signal; in the other, the mechanism is responding to it. Given a signal, to find production pitch one has to undo certain transformations to recover the parameters responsible for it. For reception pitch, one has to simulate transformations.

For us, the primary examples of this kind of pitch are the various definitions of psychoacoustic pitch: the percept which is invoked to explain the human ability to match tones of widely varying spectral distribution, to recognize melodies played on different instruments, etc.

There are good overviews of the psychology and physiology of human pitch perception in the literature ([86], [87], [65], [84], and the references cited there), and I will not assay one here. However, I would like to outline enough of the phenomena of pitch perception to convey the complexity of the field, and to demonstrate that psychoacoustic pitch is not the same thing as mathematical pitch.\(^{15}\) Probably the most renowned effect in the field is that of residue pitch: for certain multi-tonal stimuli,

\(^{15}\)Much more information on all these phenomena can be found in the references just mentioned.
the tones are not heard individually, but instead a lower frequency pitch is heard. In some cases the residue pitch is the fundamental of the stimulus, whether or not a harmonic at the fundamental happens to be present in the stimulus. However, the residue pitch does not always correspond to the fundamental; for example, a classic series of experiments by Flanagan and Guttman ([33], [34], [42]) with mixed-polarity, constant amplitude, constant-rate pulse trains, demonstrated that depending on the way in which the polarities were mixed, on the pulse rate, and on the use of high-pass filtering, as many as four different cues might determine the perceived pitch: (1) the pulse rate, (2) the fundamental, (3) the frequency of the lowest harmonic, and (4) the harmonic difference frequency. Ritsma ([93]) likewise found that for some periodic stimuli consisting of only odd harmonics, the perceived pitch was the difference frequency, not the fundamental. Another well-known phenomenon in pitch perception is the "first effect" of pitch shift: when an offset is added to the frequencies of the harmonics of a broad-band periodic signal, making the tones become inharmonic, a pitch is still often heard, at a frequency shifted slightly from the perceived pitch of the original periodic signal. Thus a pitch is sometimes perceived for aperiodic but tonal stimuli. The amplitudes of the harmonics can cause slight shifts in the perceived pitch. Changing the phase of the harmonics can radically change the perceived pitch. Two tone stimuli with frequencies \( f_1 \) and \( f_2 \) can produce combination tones with frequencies \( f_2 - f_1 \) and \( 2f_1 - f_2 \) (\( f_1 < f_2 \)). There is a pitch percept only for some periodic stimuli. Harmonics are only detected in a certain frequency range, and their importance to the computation of perceived pitch seems to vary with frequency. Harmonics with amplitudes relatively less than others nearby are masked out; those with amplitudes less than an absolute frequency dependent threshold are never heard. There is uncertainty and noise in the perceptual determination of harmonic frequencies and of pitch.

There are many competing theories to explain these phenomena, and many hypothesized physiological mechanisms. The field is still in its infancy. So far, the experiments have naturally and necessarily used extremely simplified stimuli—tone bursts and the like—in the interests of easing the subject's task and of isolating a phenomenon from extraneous issues as much as possible. As we learn more about how more complicated signals are perceived, no doubt even more radical divergences between perceived pitch and mathematical pitch will emerge.

It is true that for a fairly large set of simple tonal stimuli, the perceived pitch and the fundamental are very closely related—although even then it is certainly not a monotonic transformation of scales, since different signals with the same fundamental can sound as if they have slightly different pitches. To the degree to which they are similar, the efforts to understand mathematical pitch and psychoacoustic pitch can directly feed each other; however, there can be no doubt that the two are different, as we have seen above. If one considers the sound coming out of a clarinet, for example, there is a useful concept of local periodicity in the signal which is completely divorced from the idiosyncratic facts of human pitch perception: shift effects, masking, frequency resolution, and so on. In fact, one might go so far as to say that the interesting aspects of perceived pitch are not found in what it has in common with mathematical
pitch, but in the places that they differ.
1.2 What is F0?

The discussion so far has covered most of the entities that are ever associated with pitch. However, there has been one noticeable omission, variously referred to as “f0,” “laryngeal frequency,” and “voice pitch.” This is commonly conceived of as a function from time into frequency (i.e., giving a “pitch contour”), and as a quantity which is special to speech. It is usually thought of as being a low-level characterization of the vocal source of speech. However, none of the definitions of production pitch given earlier could qualify for being f0, because they are not represented by frequency values.

We do have a strong intuition that f0 exists. Two common (usually tacit) definitions of f0 are the following: 1) f0 is the same as what a correlation pitch tracker (or some other short-term analysis pitch tracker) tries to determine, and 2) f0, at any time t, is equal to \(1/t_0\), where t0 is the time interval between the glottal closures (or some other systematic point in the pitch cycle) straddling time t. In the next two sections, 1.2.1 and 1.2.2, I will argue that these definitions are incorrect: upon examination they turn out to be inconsistent with rather strong intuitions about the way the quantity should behave. Of course a person can use “f0” to mean anything he likes, and it may refer to a well-defined quantity, but that quantity doesn’t necessarily have any interesting relationship to the speech signal. I believe that the word “f0” should be reserved to refer to a third entity, the one described in Section 1.2.3. There, f0 is defined as the time varying repetition rate of actions at the larynx, and a method is given for computing f0 contours from the glottal closure location definition of production pitch.

1.2.1 It isn’t Mathematical Pitch

The essence of the first proposal, though it is not ever expressed using the categories of Section 1.1, is that f0 is the same as mathematical pitch. As we saw in Section 1.1.2.1, the driving motivation for the development of short-term analysis pitch trackers is mathematical pitch, which is independent of any model of the source of the signal, and works by simply trying to find a local estimate of the periodicity in the signal. Already we should be suspicious, because mathematical pitch works just as well for musical instruments, and f0 is usually considered to be something special to speech. To repeat: one could certainly make an identity between f0 and mathematical pitch by fiat, but that still leaves a useful unnamed quantity unexplicated.

Because this definition of f0 (as being equal to mathematical pitch) is not based on anything having to do with speech production, necessarily it gives pitch contours which are at odds with the expectations one has for any production-related definition of f0. Sometimes people attempt to explain the problems with this notion of f0 by pointing out that these pitch trackers are making an assumption that the signal is periodic within an analysis window, or that they return only an average of the period over a region. But those excuses have to do with the difficulties of implementation of mathematical pitch; one can well imagine that they would disappear in a limiting
case. The basic problem here lies deeper, and is very simple: mathematical pitch is not the same thing as \( f_0 \). There are many ways in which the discrepancies between mathematical pitch and \( f_0 \) show up:

- Because a mathematical pitch tracker operates blindly, irrespective of any knowledge special to speech signals, it of course isn’t aware of the distinction we make between voiced and unvoiced sounds. Because there should be (non-zero) \( f_0 \) at all and only the voiced regions of a signal, when these pitch trackers are used, they must be augmented by a “voiced-unvoiced decision” maker, which is based either on the state of the tracker itself, or is totally external. It would seem unlikely that a properly general definition of the notion of local periodicity/aperiodicity would capture all the idiosyncrasies of the voiced-unvoiced distinction—indeed, if it did, it would be a poor choice for such a definition.

- It is well known that often speakers are capable of going into a phonatory mode in which alternate pitch cycles are more similar than successive ones; this is sometimes referred to as paired pulsing. Any implementation of a good model of mathematical pitch will give a pitch value for that type of speech which is roughly twice of what the “\( f_0 \)” (something like glottal vibration rate) is; again, if in fact it doesn’t respond this way, then it can’t be a very good model.\(^{16}\)

- It would seem that any definition of \( f_0 \) should be able to say something about an isolated glottal pulse surrounded by unvoiced speech, as might occur in an extremely reduced schwa. Yet this definition cannot distinguish it from a totally voiceless region.

- Neither vocal fry, where successive period lengths vary a great deal in an apparently random way, nor speech with marked and frequent segment transitions, where successive pitch periods in the signal are very different, would appear to be quasi-periodic.\(^{17}\) But they should have an \( f_0 \).

At the root of all these examples is the fact that a pitch tracker of mathematical pitch returns a description of the signal itself, whereas a pitch tracker of production pitch returns a description of how the signal was made. Note that the characterizations of

\(^{16}\)Paired pulsing presents a potential problem for a definition of \( f_0 \) as well, when there occur cases that flout the assumption that something either is a closure or it isn’t, with no continuum between the two discrete extremes. But that ambiguity is of an entirely different order from the problem here. This is also different from the problem of correlation discussed in Section 1.1.2.1.

\(^{17}\)Those claiming otherwise are probably basing their intuition on the existence of those repeating pitch cycles in the signal. But that is exactly my point: \( f_0 \) must defined in terms of higher-level entities, and mathematical pitch cannot do that, because it is phrased in terms of the extrinsic properties of the signal.
production pitch given earlier in Section 1.1.1 have no problem with representing any of these phenomena.

An aside: the issue we have been discussing is not whether mathematical pitch and production pitch are identical, but whether the first can be subsumed under the second. The definitions of production pitch which are at a higher level than the glottal closure time characterization have nothing to do with mathematical pitch. It was argued in Section 1.1.3 that mathematical pitch is different from psychoacoustic (reception) pitch. Except for the dual role played by intonation in speech communication, it should be clear that production pitch and reception pitch are distinct, if one accepts the claim that both of them are distinct from mathematical pitch. So the effect of this section is to establish that there are three distinct categories of pitch definitions. In Sections 1.3.2 and 1.3.3 we will consider some possible definitions of pitch which do not fit into any of these categories, and we will try to show that they are not reasonable (useful, informative) definitions. Thus I believe that it is safe to conclude that production pitch, mathematical pitch, and reception pitch constitute a partition of all reasonable definitions of pitch.

1.2.2 It isn’t 1/T0

The second of the common definitions of f0 is better than the first, in that it is based on the signal’s underlying structure. But it also has problems; it results in a quantity which is much different from what one would expect from a “rate” parameter. The f0 that this definition determines is absolutely constant between glottal closures, and then has discontinuous, instantaneous jumps at the closure times. Moreover, like the preceding definition, it has a problem with voicing; during an unvoiced portion of speech, it gives a small but non-zero value for the f0 there, because it simply takes the reciprocal of the time interval between the last glottal cycle before the unvoiced region, and the first glottal cycle after it. Pitch trackers which use this definition generally solve this difficulty by either using a v/uv decision maker to zero out those unvoiced portions, or by fixing an arbitrary quantity, say T_{max}, which is an artificial hard threshold on the maximum inter-closure time distance that will get a non-zero f0. Such an ad hoc procedure is undesirable as a definition of f0. This definition also cannot represent isolated glottal pulses very well, and when the successive inter-closure times change quickly (as in segmental effects), the 1/t0 definition is close to meaningless.

To repeat the point I made when criticizing short-term analysis pitch detectors: these definitions might not be so bad as implementations; they might give f0 contours which at least visually look quite realistic. But before we accept them we should determine exactly what the ideal f0 contour is, so that we can then make a more intelligent choice of compromises in its implementation.
1.2.3 F0: Converting Events to Rate

The purpose of this section is to develop a definition of f0 which accords with the intuitions causing us to reject the previous two proposals. It would appear from the criticisms we made of those proposals that what we want is something special to speech (that is, a kind of production pitch). In particular, f0 should be the repetition rate of some kind of significant event at the larynx; glottal closures are the most obvious choice.

In working out how this repetition rate should be defined, it is helpful to realize that this is actually an instance of an extremely common situation. The problem of determining a time-varying rate function, given the moments of particular recurring events—in this case, glottal closures—comes up time and time again. The same problem could be posed for typing, clapping, or neural firing. A particularly useful analogy for glottal closures is the situation of a person practicing her tennis strokes by hitting a tennis ball against a wall. The two have many things in common: (1) in both cases, we have a very strong feeling of a repetition rate, and yet no actions are ever exactly repeated; (2) there is no precisely definable moment in the action that can be called the instant of repetition (closure or hit); (3) there is the problem that by representing all hits (closures) by a single event, we are conflating several phenomena (forehands and backhands, for example); (4) there isn’t always a yes/no answer to whether the particular event occurred (double hit, partial closure); and (5) the rate has to do with only hitting (closing); the backswings (openings) can occur at any time in between, and have nothing to do with the rate function.

The representation of the time-varying rate associated with repeating events is not difficult to find. Suppose one had two idealistic devices: a device which flashes a light at a constant rate, exactly once every time unit, and a transparent box with a knob on top which determined the ratio between the rate at which time passes on the outside, to the rate on the inside. Then by putting the first device inside the second one, and turning the knob back and forth, we can make the box flash at any sequence of moments we like. The knob could be held at a value of 1, and the flashes will occur regularly every time unit. It could be moved up to 3, and the flashes will occur three times as fast. Or it could be brought down to 0, and no flashes at all will occur. The pitch contour (f0, rate function) corresponding to the flash times is the position of the knob as a function of time.\(^\text{10}\)

This could be considered to be a special case of the definition of quasi-periodicity given in 1.3. Here \(g(\tau)\) is the periodic sequence of events, on the \(\tau\) time axis; \(s(t)\) is the given aperiodic sequence of events, on the \(t\) axis; and \(F\) specifies a curve which

\(^{10}\text{Note that this could be reversed: a pitch contour for some repeating events on the outside of the box is the knob position as a function of time which is necessary to make the events appear to happen at a constant unit rate, when viewed from inside the box. Note also that this model suggests the importance of special relativity in pitch determination, an aspect that has been surprisingly and woefully neglected in the literature.}\)
passes through all the pairs \((t, \tau)\) where events on the two axes correspond. The slope (derivative) of \(F(t)\) is the pitch contour—the knob position function \(f(t)\). Also, of course, integrating the pitch contour \(f(t)\) will obtain \(F(t)\). The condition that \(F(t)\) be nondecreasing is equivalent to a requirement that the knob never go to a negative position, thus making time go backwards. While the dynamic time warping (DTW) algorithm ([67], [70]) is irrelevant to this problem, it uses the same representation.

It can be easily seen either from the blinking light model or from the DTW representation that this definition is very ambiguous. There are an infinite number of possible pitch contours (knob position functions) for a single event sequence: namely, the \(f(t)\)'s from all continuous nondecreasing functions \(F(t)\) going through the points of event correspondence. This is actually what you would expect; the problem is to find an estimate of the instantaneous rate for all time, when all that is given is the behavior at a countable number of event times. The contour would be completely specified if complete information were given, but this is only true for trivial problems such as determining the revolution rate of a car tire from the speed of the car. This ambiguity is unavoidable; the problem is inherently underspecified. It isn't simply a matter of a lack of knowledge; there isn't any other evidence that applies for any of the cases we have looked at: typing, tennis, firing rate, clapping, or glottal closures.

There are no assumptions here which are special to speech; the transformation will work as well and will give the same result, no matter where its input came from. The resulting \(f_0\) contour has nothing to do with any other model, and is not accountable to the requirements of any other data. In particular, higher-level notions of \(f_0\) have no role here (say, in suggesting constraints on how \(F(t)\) can behave), because here the closures are taken to be the prime determiners of rate. As we saw earlier, the higher-level definitions of production pitch do not depend on the closures, and pitch contours from them would give \(F(t)\)'s which wouldn't even pass through the correct points.

By this definition, \(f_0\) does not have any independent status, but is determined from the glottal closure time characterization (or some minor variation on that theme). The best way to think about the resulting \(f_0\) contour is that it is simply an alternative mode of presentation, or representation, of the characterization from which it is obtained; it has exactly the same information as a representation which is simply a list of closure times,\(^{20}\) but it has more immediate meaning to human beings. Although this is merely a transformation on another characterization, the transformation isn't all that direct, and I think that this description is what most people really mean by "\(f_0\)."

The upper graph of Figure 1.3 shows three different possible \(F(t)\)'s which satisfy the requirements of the definition, and the lower graph shows the \(f(t)\)'s (pitch contours, rate functions, knob position functions) corresponding to those \(F(t)\)'s. It can be seen that the \(1/t_0\) definition considered in the previous section corresponds to the \(F(t)\) formed by linear interpolation between the points \((t, \tau)\) of event correspondence. With

\(^{20}\)This isn't quite true. To reconstruct the event times from the pitch contour it is also necessary to know the time of the first event.
Figure 1.3: The conversion of events to rate. The moments of the events here are 0.0, .01, .03, .037, .055, .07, .09, and .1. In the upper graph are shown plots of three different methods of interpolating between the points $(t, \tau)$: with steps, with a straight line connection, and with a cubic spline. In the lower graph are shown the rate functions corresponding to those interpolations, calculated by differentiation. The step interpolation produces impulses, the linear interpolation produces plateaus, and the cubic spline produces a smooth piecewise parabolic curve.
this representation, it is clear that the problems with the 1/t0 idea—no good response to an isolated pulse, constant values and jumps, non-zero value in voiceless regions, and bogus values when pulses change quickly—can be solved with some other member(s) of the F(t) family.

It is not clear, however, exactly what the conditions on F(t) should be, to make it more compatible with our intuitions. It would seem that we want some kind of smoothness constraint; three possible ways to formalize this are: (1) minimize the magnitude of the higher derivatives of F(t), (2) take the F(t) with the lowest high frequency energy (this is suggested by the potential for aliasing in sampling f(t)), and (3) choose the F(t) so that s(t) = g(F(t)) is minimally different from s1(t) = g(F(t) − τ1) for small τ1 (i.e. less than g(t)'s period). These are all very similar; in fact in some instantiations they would be identical. It should be recognized that these are entirely arbitrary constraints, and have nothing to do with the phenomena themselves. It is difficult to keep ideas about higher-level f0 separated from these constraints, but as mentioned before, this must be done because the F(t) from higher-level f0 wouldn't even go through the points (t, τ). Unfortunately, our desiderata for the proper interpolating F(t) are probably hopelessly capricious—someone might want the pitch contour to fall instantaneously at the onset of voicelessness, and yet not want any such jumps during voicing—so it is probably a futile enterprise to try to formalize the constraints.

As shown in Figure 1.3, a unit amplitude pulse train is a pitch contour by this definition, so that any smoothing action on F(t) can be seen as a kind of smoothing of that impulse train. In fact, when the rate isn't changing significantly over a few quasi-periods, the result of convolving that impulse train with a tapered mask of unit area and a width of a few quasi-periods will give a (visually) acceptable pitch contour. Unfortunately, the only convolution mask which is guaranteed to provide an F(t) going through the right points (t, τ) is the impulse δ(0), which is hardly useful. The only way that a valid F(t) can be obtained from the impulse train is from nonlinear processing, as is used in the 1/t0 definition.

In light of these observations (the weakness and inconsistency of our intuitions, and the lack of any obvious way to simply get a smooth contour), the 1/t0 definition may be the best way to implement f0 for applications. If the contour is just for visual presentation, or if f0 isn’t changing quickly, it should certainly be all right. Otherwise, some improvement might be obtained by using the derivative of a smooth spline through the points (t, τ).
1.3 The Three Confounding Factors

We started this chapter with a brief outline of three things which hinder effective study of pitch, to which I gave the labels "terminology," "implementation," and "application." Now that a framework has been established and some definitions of pitch have been given, in the next three sections I would like to expand on the problems caused by these three factors.

1.3.1 Terminology

"But 'glory' doesn't mean 'a nice knockdown argument,'" Alice objected.
"When I use a word," Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean—neither more nor less."
"The question is," said Alice, "whether you can make words mean so many different things."
"The question is," said Humpty Dumpty, "which is to be master—that's all."

Lewis Carrol, *Through the Looking Glass*, 1872

It is a truism that there is no and can be no best pitch tracker. One can always make a pitch tracker which works faster at the sacrifice of accuracy, for example; it is up to the designer to choose the particular characteristics he wants. This is an extremely common phenomenon. It is much more important and interesting to note that there is no single best definition of pitch. Each of the different interpretations of "pitch" I have been describing has a completely different collection of data, folklore, knowledge, intuitions, etc. to be dealt with. The fact that they are different doesn’t mean that only one of them can be "right"; it means that they aren’t rivals at all, because they are talking about different things. They deal with different data, so of course they will give different results, but that doesn’t mean they are inconsistent; it only means they will be useful in different circumstances. Further study won’t later cause one of them to be later labeled as wrong, though it might change their relative utility. Unlike the engineering reason for there being no best pitch tracker, this multiplicity is not something to bemoan, because it really doesn’t cause any inconvenience, but it must be recognized and accepted. It is neither a compromise nor a simplification to study only one specific interpretation of pitch; there is no deep interconnection between the different interpretations that one can investigate.

Thus concern about the "best" or "correct" meaning of the word "pitch" falls properly within the domain of prescriptive lexicography, and has no bearing on the question of what is a suitable subject of research. All three of the categories are
different, and all three are useful (see Section 1.4.2), no matter what one wants to call them. The fact that psychoacoustic pitch has some claim over the word “pitch,” for example, surely does not mean that all algorithms in the field should be striving for psychoacoustic pitch. Similarly, the feeling that everything must be represented as a pitch contour (a function from time into frequency) is basically a pre-theoretic bias which is unjustified for many of the notions of pitch we have examined—including psychoacoustic pitch (see Section 1.4.2.3). If someone is of the opinion that pitch is no longer being talked about when these conventions are violated, then fine, it can be called something else. The interesting and substantive issue, which is untouched by this caviling, is which of the two is more appropriate for describing the particular phenomenon: “real” pitch, or the definition/representation that is being denigrated.

I have been using, and will continue to use, the word “pitch” to refer generically to all of the entities so far described, and to any other interpretation someone might propose. Some word has to fill this role, and “pitch” is the obvious choice. Puritans will insist that the only proper use of the word “pitch” is as a synonym for “psychoacoustic pitch,” which they regard as redundant. I think the ambiguity is here to stay, so we might as well take advantage of it; practically all researchers on “pitch” proper now put some qualifier on the word. I use the terms “pitch tracking,” “pitch determination,” “pitch analysis,” “pitch detection,” “pitch extraction,” and “pitch estimation” interchangeably to refer to the process of applying the definition/theory of one of these kinds of pitch to a signal to get the appropriate representation of that pitch. That representation may or may not take the form of the traditional time-frequency plot; when it does I will refer to it as a “pitch track” or “pitch contour,” or possibly “f0 contour” if I’m considering production pitch.

1.3.2 Implementation

It is customary to organize discussions of the past work on pitch tracking according to gross characterizations of the algorithms used. The artificiality of this division by algorithmic features is evidenced by the great variety of ways it has been done in the literature. Hess, for example, splits up pitch trackers into “Time-domain” and “Short-term Analysis.” Time-domain trackers are further split into inverse filtering and structural analysis methods, and short-term analysis trackers are split into frequency domain (including cepstrum), correlation-like methods, and maximum likelihood ([46], p.472ff). McKinney groups correlation with the time-domain methods ([63]). Sreenivas splits them up into five methodologies: correlation, cepstrum, spectral features, maximum likelihood, and pattern recognition ([113]). Baronin splits them into frequency domain, structural analysis, correlation (this includes cepstrum), and inverse filtering.21 This variation is to be expected; it is often easy to interconvert between time and frequency, and operations in one domain can be reconsidered in the

21Baronin has written extensively on pitch detection, but it is all in Russian. I am dependent on Hess's excerpts from Baronin's work. This particular portion is in [46], p.478ff.
other. Also, it would seem pretty clear that the classification of algorithms resists the imposition of a tree structure; correlation and cepstrum, for example, are an odd mix of both time and frequency.

This practice of organizing by algorithm is symptomatic of a general habit to conflate definitions and theories with their implementations—a habit I would like to avoid. That is part of the reason this chapter is mostly concerned only with pitch, and not pitch trackers; issues of implementation are not considered until the next chapter. In general, the distinction between definition (pitch) and implementation (pitch determination) permeates this thesis. Without it, for example, we could not have criticized correlation as a definition of quasi-periodicity, and still have maintained that correlation could potentially be part of a good pitch tracker (for some inputs). Likewise, we couldn’t have claimed that the first two definitions of f0 were wrong, and yet still have admitted that they might be acceptable in practice. Definitions provide an ideal; by being consciously aware of the perfect goal, one is able to make more intelligent compromises in its implementation.

It would be preferable to organize discussions of past work not by the pitch trackers themselves, but by the definitions of “pitch” corresponding to those pitch trackers. There certainly is a definition of pitch corresponding to any pitch tracker, independent of that pitch tracker, because otherwise pitch trackers could not ever be considered wrong. Conversely, a single pitch tracker can’t possibly hope to satisfy distinct definitions, so there is at most one definition of pitch for each tracker. This approach would of course be pointless if all pitch trackers shared a common interpretation of what “pitch” means, or if they all had very different ones, but because this is not so, a partition of pitch trackers according to superficial algorithmic properties could easily confound interesting typological distinctions. If implementations determine definitions, then it makes no sense to say, for example, that a correlation pitch tracker makes errors, and it is impossible to express the fact that correlation and AMDF algorithms have a common goal which they don’t share with some other algorithms. Once pitch trackers are differentiated according to their definitions of pitch, it is then possible to consider a further classification by algorithmic properties.

In the remainder of this section I hope to sharpen our understanding of this issue by placing pitch determination within a more general view of the role of implementations in science and engineering.

Three concepts which are fundamental to this understanding are data, theory, and implementation. In a nutshell, the theory (just definitions, in the simplest case) is the explanation, the data are what are being explained, and the implementation is the means by which the theory is used and tested on the data.

Each of the interpretations of pitch we have been considering has its own set of data. For production pitch, the most directly important data consists of stereo recordings of acoustic speech waves and the output of some device giving more ex-

---

Note that this use of the word “data” is very different from its use in the phrase “speech data,” meaning the speech signal.
plicit information about the actions of the larynx: an electroglottograph, a throat accelerometer, a fibroscope, etc. But the data also includes all our knowledge of the physiology, acoustics, and mechanics of speech production. For mathematical pitch, the data is ultimately our intuitions about the way mathematical pitch should behave. For psychoacoustic pitch, the data is made up of the results of the many perceptual experiments that have been done, our knowledge of the physiology and biomechanics of the ear and brain, and so on. Though the particular subset of the data that I have seen, or that any other human being has seen, is finite, the entirety of the data is infinite, because it includes all the potential recordings and experiments that haven’t been (and may never be) done, and all the other future knowledge we may (or may not) acquire about the subject. It is the job of the theory—which is necessarily finite, because people have to think them up—to summarize, explain, or give structure to the data. Theories are responsible to all of the data, and often must be changed as new information is uncovered. It is the goal of a theory to explain “as much of the data as possible.” Sometimes the theory is extremely simple; for mathematical pitch all that must be done is to come up with an appropriate definition which synthesizes our intuitions. Other times it is more complicated, as for psychoacoustic pitch. The distinction between data and theory is sometimes a little blurry; as a theory gradually moves from a mere hypothesis to almost certain knowledge, the theory may eventually be used itself as data for more general or abstract theories. Also, there are subtle philosophical issues associated with the distinctions between theories, models, and explanations. However, these somewhat superficial definitions should be sufficient for this discussion; the relationship that I want to explore in depth here is not the one between data and theory, but the one between theory and implementation.

After a researcher has decided upon a tentative theory (model) of some data that is of interest to him, he will of course want to test, and/or eventually use, his theory on the data. Theories are ethereal passive objects; there must usually be something else to mediate a theory’s connection to the data. By “implementation” I mean whatever makes this test/use possible—the avatar of the theory. While this concept is useful when computers are not involved, it becomes particularly important when a theory is implemented as a computer program, and that is the case that I will concentrate on most.

There are four main reasons that implementations are not isomorphic to the theories to which they correspond.

1. There is the possibility of “bugs” in the implementation of a theory: if a computer is being used, then there will be inevitable initial programming mistakes; if testing is being done by hand, the numbers may be added together incorrectly, figures might not be copied accurately, etc. There can be unknown discrepancies both between what the procedure does and the intentions of the implementor, and between the intentions of the implementor and what the theory dictates. Bugs certainly occur in real life and cause a lot of grief and wasted time, but they are totally irrelevant and uninteresting here, and I will ignore them from now on.
2. There is a lot more to an implementation than the aspects which have to do with the theory. The researcher has to make a host of decisions about things which constitute necessary attributes of the implementation, but which are completely unrelated to the theory. If he is doing his implementation on a computer, this category includes such choices as computer system, programming language, data structures, modularity of the code, etc. (If he is doing it without a computer, this would include such things as the kind of writing utensil he will use.)

3. The exact statements of a theory can be prohibitively difficult, and in the interest of getting some idea of how well everything is working, a researcher will make some approximation to what the theory actually requires. This could be necessary because the technology available is primitive, or because the specifications of the theory are intrinsically onerous, or because the researcher isn’t creative enough to think of an alternative.

4. Even in the ideal case where one can incorporate every last thing a theory has to say into its implementation, if the theory is incomplete, the implementor will have to fill in the holes, because to apply a theory to data one needs to specify every last action. Often in this case the researcher is guided (i.e. constrained) in his choices by his intuitions about plausibility. Presumably the results of experiments with an initial implementation will reveal further improvements of the theory, or at least will restrict the possibilities.

Notice that (1) can be made to be a null set; (2) is never null; (3) sometimes is null and sometimes isn’t, according to the theory; and (4) is never null for interesting problems.

It is important to recognize that the problems of (3) do not in any way reflect on the theory. No one believes that all the things one might want to study in the world are for some mysterious reason perfectly suited for simulation on a present-day computer. Theories don’t have to answer to the idiosyncrasies of a silicon finite state machine unless they happen to involve hypotheses about machine computation, and that isn’t the case here. There are many good reasons for implementing a theory on a computer. But because the theory may have qualities making it a poor match with standard computers—it may not be algorithmic, or it may involve a search through a huge space—compromises will have to be made in the theory just to be able to translate it into a program that will run in a human lifetime. This is an unfortunate fact of life, but it doesn’t mean that the theory is wrong, poor, or even limited.

Note that there definitely is a qualitative difference between (2) and (4). No reasonable (useful) definition of “completeness” of a theory would require that it specify every last aspect of its implementation. This would mean that a complete theory would have to incorporate all possible future technological innovations. To get away from computer examples, a theory of mouse-trapping, for example, might specify that a spring-loaded clamp be made of some rigid, durable material. If a person chose to use a clamp made out of glass crystal, he would be violating the dictates of his theory, and indeed his performance would be poor in theoretically relevant ways. However,
he could choose many different kinds of metals, or even synthetic materials, without violating the theory, and his performance would be identical with all of them. The particular material chosen, within the constraints of the theory, would correspond to a choice in (2).

The decisions involved in (2), (3), and (4) are not independent. Clearly the computer facilities available (2) will affect the particular approximations chosen (3); this in turn will have some effect on the completeness of the model (4). Let us take as an example a model of human speech production incorporating the behavior of the lungs, the larynx, the vocal tract, the articulators, etc. If a person has ambition and a supercomputer, when he implements this model he may attempt to incorporate three-dimensional flow, nonlinear effects, and all the different muscles. There will be a lot of things he won’t know, so he will have to make a lot of guesses for (4). On the other hand, if he only had a mini-computer (2), he would have to make many more approximations in his model (3), but he would have to make fewer choices in filling in the unknown aspects (4). Conversely, the reason that the researcher may not know something (4), is that his resources (2) have always been limited.

In summary, using the terms of the preceding discussion, a pitch tracker constitutes neither data nor a theory; it is an implementation.

Pitch trackers are not data; there is always external knowledge which a pitch tracker has to answer to, though this knowledge may change from tracker to tracker. A definition of pitch which says, essentially, that “pitch is whatever my pitch tracker returns,” is not doing any work. Such a definition is clearly inconsistent with the strong intuitions one has about when a pitch track is correct and when not, because that pitch tracker is never wrong, by its very nature.

Pitch trackers aren’t theories or models. Instead of considering each of the myriad pitch trackers to be a direct and accurate implementation of a different model which happens occasionally to be grossly wrong, it is better to consider it to be an approximation to a model: whether the author (or anyone, for that matter) is aware of that model or not. Several pitch trackers, with widely varying algorithms and performance, could possibly all be based on the same model, but consist of different approximations to it.

In light of these latest observations about the differences between models and implementations, it should be clear that a more profitable way to look at pitch trackers (and computer programs in general) is as implementations of models. What a pitch tracker does, then, is the following. It takes as its input a signal, or group of signals, which can be measured in the real world; usually this is a speech signal. Its output is a characterization of this input, according to the dictates of some theory; this characterization is of course only partial (it doesn’t allow reconstruction of the input), and it tells something about the “pitch” of the input. Thus a pitch tracker crucially depends on some theory, because its output is only useful, and can only be interpreted, theory-externally; indeed, it may happen that the output signal cannot be measured at all in the real world, and has consequence only as a theoretical construct.
1.3.3 Application

The reasons for implementing a theory fall roughly into two categories: (1) to help improve the theory (by facilitating its testing, for example), or (2) to apply the theory in a particular area (to meet a commercial demand, for example). This is the same as the test/use distinction alluded to earlier, and, one might argue, it is the crucial factor differentiating Science and Engineering (but perhaps not scientists and engineers).

This difference in objectives will obviously have a big effect on how the implementation is done and on how it is evaluated. In the scientific case, ideally the foremost (and only) criterion for choosing an implementation is that it represent the most efficient path towards the attainment of truth. In making the decisions associated with (2)-(4) above, the Scientist must consider the tradeoffs inherent in the requirements that it not be a very onerous programming chore, that the implementation not use an unreasonable amount of time, space, or money, that the approximations not be too egregious, and that she not have to guess about too many things. The goal is to obtain an implementation which is as true to the theory as possible, where “trueness” is measured in roughly the same way that accuracy is measured between the theory and the data, and “as possible” is determined by the constraints just listed.

The Engineer is presented with the same array of choices, but he has additional concerns, and the ones they have in common he will weigh differently. For example, he may have to have real-time response, or he might have to keep the weight of the final device under 20 lbs. From a scientific point of view, the different implementations arising from different choices in (2) form an equivalence class: A DFT is a DFT, whether it is implemented with Lisp or Fortran, and it doesn’t matter what DFT algorithm is used. But in an application those questions are of great interest. Most of the properties of an implementation that are important in an application have nothing to do with the model behind the implementation: cost, weight, memory, speed, etc.; in fact, the only one that does is accuracy.

The two different possible motivations for an implementation lead to different conceptions of accuracy. Unlike the case of a Scientist, whose single evaluation measure is accuracy, in an application accuracy is just another aspect of the implementation to be payed for; applications often use crude approximations to truth in the interest of conforming to other requirements. There is no longer an absolute scale of comparison, but instead a huge trade-off space of cost, size, accuracy, speed, etc. The need of an application for accuracy is different; in some cases, an application really wouldn’t benefit from further accuracy, either between the implementation and the theory, or between the theory and the data. In other cases, the theory is simply not good enough, and so the application cannot be done. Moreover, accuracy might be measured in much different ways by the two: discrepancies from the data that might not bother an application at all might have significant consequences for a theory, and it is possible that errors measured, say, with an L2 norm, that might be intolerable for the application might not mean much to the theory.

These different conceptions of accuracy can in turn lead to problems in defining
what the implementation is trying to do. A case might be made to the effect that
accuracy in an implementation for an application should be dealt with just like all the
other properties, rather than by making reference to the theory: the most accurate
pitch tracker for a vocoder is whatever makes it sound best; the most accurate pitch
tracker for a language instruction aid is whatever makes language acquisition most
efficient; the most accurate pitch tracker for speech recognition is whatever gives it
the smallest error rates, etc. What this would mean, essentially, is that there would
be new definitions of pitch created; “vocoder pitch,” for example, would correspond to
the following: “the pitch of a speech signal is whatever determines the vocoder excita-
tion producing the most intelligible (or most natural, or most like the original, etc.)
synthesized speech according to subjective listening tests (or signal comparison).” But
this definition depends as much on the idiosyncrasies (constraints and assumptions)
of a particular vocoder and on human judgment as it does on properties of the speech
signal. There doesn’t appear to be anything but engineering interest in this definition;
I see no reason that the resulting correspondence between speech and pitch should
show any order beyond the way it reflects the properties of one of the other pitch
definitions already discussed.

The problems with vocoder pitch are indicative of the problems with the general
position that pitch should be defined functionally. First of all, because the testing
criterion for the pitch trackers is based on performance in an application, it is quite
conceivable that a perfect pitch tracker by theoretical/scientific standards would not
work as well as something which has been adjusted to conform to the special situation.
Thus the best vocoder excitation may not necessarily be the one corresponding to the
original vocal source; the best contours for language learning may not necessarily be
correct pitch contours; and the best parameter to add to a speech recognition feature
vector may not be pitch. The problems of improving the characterization of the source
in a vocoder, of making the representation of pitch more perspicuous for the language
learner, and of improving the general performance of a speech recognizer, are really
entirely separate research problems, and should be treated as such. Secondly, it is
hard to imagine (at least for the above cases) how one would even go about optimizing
a pitch tracker according to only the application criterion. It would seem necessary
to incorporate at least something from the theoretical model—but then one gets into
the rather distasteful situation of trying to meet conflicting goals. Finally, it would
be expected that the differences between the application-based notion of accuracy and
the theory-based one would be unsystematic and idiosyncratic. A slight change in
the application might have a big effect on the functional definition of accuracy, and
little could be shared between applications in different areas. Particularly in this age
of exponentially increasing hardware speed and density, it would seem rather short-
sighted to be relativized to the demands and restrictions of current technology.

In closing this section, it is encouraging to note that the definitions of pitch sug-
gested by implementation (based on what the algorithm returns) and by application
(based on how the algorithm is used) are not readily amenable to categorization within
the typology presented in Section 1.1, thus giving a further example of the utility of
that typology.
1.4 Some Consequences

The basic observation that there is more than one kind of pitch is by no means novel—many grade school children learn to distinguish the words "pitch" and "frequency," and it is universally admitted in the research community as well. It is important to develop this distinction, as I have been doing so far, but it is also vital to expose the consequences these distinctions have for pitch determination, because those consequences have sometimes not been widely recognized.

The different definitions of pitch constitute the possible goals of a pitch tracker, and it should be clear that the definitions will determine, or at least strongly influence, the answers to questions about pitch tracking: questions about representation, application areas, implementation methods, errors, testing, comparison, improvement, and so on. Now that we have made explicit (although not very detailed) the underlying definitions of pitch, we are in a much better position to proceed on to an examination of these issues.

Making the pitch definitions explicit is important not only for discussion; it is of course important when developing pitch trackers as well. Because the definition determines the answers to those questions, and because different definitions give different answers, it is absolutely vital that the particular definition be known. I believe that the typology of pitch definitions given earlier is complete; that is, the only reasonable goal of a pitch tracker is to obtain one of the definitions contained in it. It is even more important, perhaps, that no pitch tracker should be involved in a mixing of more than one definition. Obtaining such a hybrid definition is of little interest, and necessarily the performance on each of the individual constituent definitions will be hurt by the association with the others. As a practical matter, inferior definitions of pitch and hybrid definitions sometimes may not differ much from each other or from better definitions, when they are implemented. But it is only by clearly identifying what it is that we are trying to attain that we can hope to do an effective job of testing, comparing, or improving implementations. It is possible that some applications don't require the accuracy that it takes to distinguish these different definitions. However, those applications are not of much concern here, because there already exist pitch trackers which satisfy those loose requirements. It should be clear that to get anything approaching highly accurate performance, it is necessary to specify precisely what kind of pitch is being modeled.

Examples of pitch trackers which involve more than one kind of pitch are not hard to find. Many correlation and AMDF pitch trackers, which have their foundation in mathematical pitch, are applied not to the speech signal directly, but to a linear prediction coding (LPC) residual derived from the speech; this preprocessing helps because the transfer function of the vocal tract is often all-pole. Thus information special to speech is used—a hallmark of production pitch. Similarly, cepstrum pitch trackers are also basically associated with mathematical pitch, but they can only be expected to work on speech-like signals. At present, such approaches might be necessary. However, this path is ultimately a dead end; it will be necessary—soon I think—to choose
between mathematical pitch and production pitch. These hybrid trackers are limited in their usefulness for mathematical pitch by the fact that their performance is dependent on the input being like a speech signal. And because the basis of the algorithms is mathematical pitch, even when aided by properties special to speech signals, they still will be incapable of doing really well at getting production pitch ($f_0$, for example). The problems with using mathematical pitch for $f_0$ are not going to be eliminated by pre-processing or post-processing; the solution is to recognize the faults in the basic assumptions about $f_0$. I believe that the better pitch trackers of mathematical pitch already work almost as well as they can be expected to on speech; the problem with them isn't so much that they are wrong, but that they are based on a model which is inappropriate and therefore sometimes useless.

1.4.1 Input Constraints

The choice of a definition of pitch and a particular set of phenomena to model naturally leads to statements about the form of the output of a pitch tracker for that kind of pitch. The choice also has consequences for the input to the pitch tracker. The theory/model/definition which a pitch tracker implements usually concerns itself with only a particular subset of the space of all signals. Thus, for example, the theory behind a tracker of production pitch is only about how speech signals are produced, so that it is implicitly assumed that the input to such a pitch tracker is indeed a speech signal. The theory says nothing about other possible effects on the signal as it is actually measured, such as phase distortions, background noise or rumble, rectification, bandwidth limitation, polarity reversal, etc. A tracker of the intonation kind of production pitch is even more constrained, because there the theory is specific to a single language. Similarly, trackers of mathematical pitch or reception pitch will also not normally have the capacity to deal gracefully with signals falling outside the domain of the phenomena being modeled. If, for example, it is decided that psychoacoustic pitch for speech is significantly different than that for other sounds, it will be necessary to restrict the inputs accordingly. All of these constraints are perfectly reasonable—one would never expect a parser of English to be able to handle Japanese or Fortran, even though those other inputs fall into the same space of possible symbol sequences. Moreover, there is no all-encompassing system to these distortions and bad inputs which would allow one to incorporate them into a pitch tracker at once; they are just a disorganized set of different noise sources having no common properties, so that each would have to be considered individually and adjusted for in a pitch tracker. A straight implementation of a theory which had no consideration of these other inputs might well fail miserably if one of them were present. This is obviously of no consequence for the theory, because it wasn't intended to be able to handle this kind of input.

Equally as obvious, however, is the fact that this kind of behavior might make a sensitive pitch tracker unpopular for some applications. Sometimes a pitch tracker
is summarily dismissed because it falls apart on synthetic speech, or speech with its polarity reversed, or telephone speech, or infinitely peak-clipped speech, or bird calls, or foghorns. Some of the time this is justified; i.e., the phenomenon under question does fall under the jurisdiction of the theory being implemented and therefore the pitch tracker should be able to handle it. Other times, even though the particular type of signal isn’t strictly legal, it happens to be so important for the application that special adjustments are merited. But frequently these complaints are motivated by arguments are confused about what the goal of the pitch tracker is. A person might maintain, for example, that because humans can understand infinitely peak-clipped speech, any pitch tracker should also be able to deal with it: this is only true for pitch trackers of psychoacoustic pitch, if it is true for any at all. Likewise, when people state that any algorithm for finding periodicity should be able to deal with polarity reversals and phase distortions, or it isn’t any good, they are probably using intuitions about mathematical or psychoacoustic pitch that are inappropriate for production pitch. A tracker of production pitch can’t be and shouldn’t be consistent with psychoacoustic facts.

Note that there is still much variability that a pitch tracker may have to deal with, even among its proper inputs. A tracker of production pitch, for example, might be flexible enough to be able to adjust to healthy speakers of both sexes and of all ages. But unlike the various kinds of distortion above, there are some systematic common properties to all of these inputs, and the variation is somewhat parametric. It is doubtful that a pitch tracker which was able to accommodate this variation successfully would actually fall apart on inputs which were not speech but were speech-like, but again, this ability, and the rate of its degradation, is of no theoretical importance.

The issue of input constraints cuts both ways; if a definition of pitch does not rule out a particular input, then the implementation shouldn’t either. Mathematical pitch shouldn’t be restricted to speech, and few current proposals for psychoacoustic pitch are specific to speech either, so the phrase “pitch determination of speech signals” is only appropriate for production pitch.

Consider the glottal closure location kind of production pitch. This kind of pitch only makes sense for a restricted class of signals; the concept of a beginning of a period is meaningless for an arbitrary periodic signal. You wouldn’t expect this kind of pitch tracker to work on environmental sounds, such as bells—how could it? And the only way the pitch tracker could work on speech signals is if it had built-in knowledge about how speech is produced. Thus:

\[ \text{It is possible to find out information special to a signal if and only if} \]
\[ \text{assumptions special to the signal are made if and only if} \]
\[ \text{the input is restricted.} \]

You cannot have your cake and eat it too. A claim that a pitch-tracking algorithm will work about as well with speech or LPC residual as its input, or that it will work on both speech and music, is a sure sign that whatever the algorithm is returning, it can’t be production pitch.
1.4.2 Applications

It would have been preferable to discuss applications earlier, as part of a general introduction to the subject of this thesis. However, there was good reason to postpone it. As was noted in Section 1.3.2, it is important that pitch trackers be organized according to their definitions of pitch, before their algorithmic properties are considered. The same thing applies to discussions of the applications of pitch trackers: applications are best considered first on the basis of the type of pitch definition used, and then on the basis of algorithmic technique. I didn’t feel comfortable talking about applications until the typology of Section 1.1 had been established.

I would now like to give an overview of the current application areas, and to indicate which of the various kinds of pitch described so far is most appropriate for each of them. Because some of the kinds of pitch are similar, often more than one of them could conceivably be used for an application, but there is nearly always one which is the natural and preferable choice. Here I am restricting myself to just the instances of production pitch having to do with speech production, and instances of reception pitch having to do with human pitch perception.

1.4.2.1 Production Pitch

Production pitch is currently by far the most useful, and it is difficult to imagine that this will change in the future. It was argued in Section 1.1.1 that the glottal closure representation is a good description of low-level production pitch. But there are some cases for which a representation as a time-frequency curve is more useful. I don’t think anyone would disagree with the claim that time-frequency curves are a more perspicuous mode of presentation, so for any application involving humans—for example, some linguistics research and speech therapy—it would be the desired final product. In the list below, I have tried to indicate whether glottal closure locations or \( f_0 \) is preferable.

Some of the applications of production pitch are:

- Speech analysis and speech processing.
  To a rough approximation, voiced sonorant speech is the response of a passive slowly time-variant linear system, corresponding to the vocal tract, to an impulse train, corresponding to the puffs of air from the oscillating glottal folds. Because most of the energy in the source excitation is concentrated temporally close to the glottal closure times, spectral estimates for the linear system can be made much more precise if they can be done pitch-synchronously. Many kinds of signal processing procedures depend on a short-term analysis window; the results of all of them can be made more accurate and revealing if it is possible to more intelligently select the window location and width. While some improvement may be obtained from just knowing the local period length, it is better to also know where the pitch periods begin; so glottal closure locations would be preferred over \( f_0 \). Having this information also allows editing of the signal (insertions or
deletions) to occur on cycle boundaries, which would, for example, facilitate the preparation of high quality stimuli for speech perception experiments. This also is slightly easier with closure locations than with \( f_0 \).

- **Speech synthesis and speech coding.**
  Knowledge of glottal closure locations is vital for vocoders and linear predictive synthesizers; in fact, if all those algorithms are given is an \( f_0 \) contour (whatever that is), they have to convert it into an approximation of the characterization anyway. Because of the non-uniform distribution of energy in the voiced excitation source, knowledge of the closure times would be helpful to the many speech coding techniques which depend on summarizing a prediction residual, such as multi-pulse LPC.

- **Linguistics.**
  For any study of phonetic voicing, of voice-onset times, of glottal stops, or of the segmental effects on production pitch, to name a few examples, a dependable representation of where glottal closures occur is vital. As we saw in Section 1.1.1.1, all low-level interpretations of voicing are dubious, but of them, one in terms of closures seems most reasonable. Pitch contours as they are usually found are close to meaningless where the pitch changes quickly; so for segmental effects closure locations are clearly preferable over \( f_0 \). Moreover, an ambitious study of segmental effects would want to investigate the importance of the timing relation between glottal closures and the asynchronous segmental effect (burst, nasal release, etc.). That cannot be done with an \( f_0 \) contour. Pitch trackers are of course also a necessary tool for studies of the role of \( f_0 \) (or closure times) in intonation.

- **Speech recognition.**
  In addition to improving low-level signal processing, a production pitch tracker could be useful at later stages of speech recognition as well; the accurate representation of voicing and of the micromelody could aid phoneme identification, and the pitch contour could be of help in analyzing the prosody of the utterance.\(^{24}\)

- **Speech pathology.**
  The amount of jitter in the \( f_0 \) (or the glottal closure locations) is related to both

---

\(^{23}\) The connection between pitch tracking and speech recognition is not only practical. Both involve the determination of underlying information in speech signals, using knowledge about how that information is conveyed. The strong parallels between the two problems implies that the two endeavors can inform and complement each other a great deal.

\(^{24}\) It is doubtful, however, that word boundaries have a reflex in English \( f_0 \) contours, as has sometimes been claimed. See, for example, [85], [55], and [68]. However, evidence for word boundaries might be derived from the stress pattern and syllable structure, and those in turn might be found with the help of \( f_0 \).
the emotional state of the speaker, and to physiological factors. Thus this could be used as indicator of stress (as in lie detection and the analysis of baby cries), or in the diagnosis of pathologies.

• Language instruction and therapy.
The feedback of seeing both the speaker's f0 contour and a paradigm is useful both for teaching the deaf their first spoken language and for teaching nonhandicapped people a second language. For speech therapy, this display might be augmented by an indication of breathiness, a voice efficiency measure, and/or an indication of jitter. This could also be useful for training singers.

1.4.2.2 Mathematical Pitch

For speech, mathematical pitch is rarely if ever the ideal interpretation of pitch to use. However, it does have some applications, both within speech research and without:

• A robust approximation to another interpretation.
It is possible that only a pitch contour is needed, that the signal is so distorted that it violates the input constraints of a pitch tracker of the ideal interpretation of pitch enough to make it unusable, and that the unavoidable fine errors from substituting mathematical pitch are not as important as the added robustness. In that case, it is reasonable to use mathematical pitch instead of production pitch or psychoacoustic pitch, when the representation of the interpretation is a pitch contour. For example, many kinds of distortion destroy any obvious structure in the waveform, even though they have little coherent effect on the short-term spectrum. Thus a tracker of production pitch which depended on simple waveform primitives might fail completely while a mathematical pitch tracker might come up with something pretty close to the right answer. While it is true that now the choice between production pitch and mathematical pitch does seem to be a trade-off between accuracy and dependability, at least for noisy speech, it seems likely that this will not always be the case, because a tracker of mathematical pitch cannot take full advantage of special knowledge about speech signals when it is trying to work around noise.

• A basis for universal description of signals.
Because mathematical pitch has nothing to do with either a signal's source or its destination, it can be used either when those are unknown, or when a person is involved with so many different signals that a universal means of comparison is desired. Mathematical pitch is nothing if not eclectic; it can be used on random environmental sounds (animal calls, instruments, engines, etc.), on artificial signals such as stock indices, or on biological signals (EEG, EKG, EEG, etc.).

• Music.
An important example of the previous case is in music; a mathematical pitch
tracker could be used in automatic transcription systems, to help tune instruments, or to give computers the ability to respond to what humans do on more traditional instruments when the two perform together in concert. Mathematical pitch is more appropriate for these applications than psychoacoustic pitch, and it is probably out of the question to develop a separate production pitch tracker for each instrument.

1.4.2.3 Reception Pitch

One of the fascinating aspects of speech research is the multiplicity of approaches to a single phenomenon. For example, in trying to determine the text of a spoken utterance, a person could attempt to reverse the process whereby it was produced, or he could attempt to duplicate the processing that goes on in speech perception. The remarkable fact is that although the intermediary stages in production between a "sentence" and an acoustic signal, and the intermediary stages in perception between the acoustic signal and the perceived sentence, are radically different (one involves the mouth and the other the ear, after all), the extrema of the two processes are similar. Thus a person working on vowel recognition could either attempt to analyze the signal into the formants corresponding to the resonances in the vocal tract, or he could transform the signal in what he considered to be a way similar to the way the auditory system does, and look for any invariance or simplicity he might gain thereby. The same is partly true for pitch determination; if one wanted to study the linguistic use of pitch, intonation, one could conceivably benefit from studying either the production or the perception type of pitch, because that aspect of the language faculty certainly plays a big part in both of them.  

At the moment, however, this point is moot, because there are no believable general psychoacoustic pitch trackers for any stage of processing, although there are some which faithfully reproduce the data on a restricted set of stimuli ([118], [119], [128], [129], [81], [39], [40], [36], [102]). For that reason, I'd like to discuss some of the reasons for the difficulty in making a psychoacoustic pitch tracker.

The basic problem is that we really don't know what such a program should return; we simply don't have the data, and we have virtually no way to predict it. This means either that it can't possibly be right or that it can't possibly be wrong, depending on your point of view. There is a tremendous blind leap from the pulse trains and tone bursts of pitch perception experiments, to speech signals with arbitrary phonetic content and intonation—and currently it is difficult to meet the observational requirements of even the data on extreme signals. Crucial aspects of pitch perception take place centrally ([50]), a stage of hearing that we have virtually no understanding of. More is required than just doing a few more experiments, however. There are two fundamental challenges to the project of constructing a psychoacoustic pitch tracker,

---

25In fact, it was a somewhat arbitrary choice on my part to include intonation with production pitch rather than with reception pitch in the typology.
which must be recognized and dealt with before it can get anywhere. First of all, it is
necessary to address the question of whether to represent the perception of the pitch
of a speech signal by a function from time into frequency. While it is possible that it
might turn out that a psychoacoustic pitch tracker should return pitch contours, it is
not at all obvious that this should be so. The pitch contour representation introduces
an entirely spurious degree of aural resolution, both in time and in frequency. People
are simply not that good at pinpointing the exact pitch of a sound, or at detecting
pitch deviations of short duration, or at localizing an event to a particular time.26 It
doesn’t make much sense to characterize a person’s response by a pitch contour when
in fact the person is demonstrably incapable of distinguishing the pitch from that of
a signal with radically different mathematical or production pitch. Also, for many of
the pitch perception experiments subjects can hear several phenomena at once, and
they have to be asked to respond only to the “dominant” pitch. In general, the results
of pitch perception experiments are often sensitive to the expectations of the listener
([36]). Efforts to model frequency masking and uncertainty by smoothing, or adding
noise to a steady-state spectrum before the pitch is extracted, fall far short; an attempt
is still being made to find a single number to summarize the response. Likewise, efforts
to model uncertainties in time resolution and time localization by simply smoothing
the pitch contour don’t even begin to address the problem.

A second question which needs to be brought out into the open is whether the
perception of the pitch of speech is a simple extension of pitch perception for impulse
trains, synthetic steady-state tones, etc. Again, it is far from clear that the affirmative
answer is the correct one. It seems hard to believe that our ability to match tones
has a simple connection to the way we interpret (decode) the intonation of a spoken
utterance. People are not bothered at all by large segmental effects on pitch or by
whispered speech; in fact, they usually aren’t even aware that anything special is
happening. Moreover, although there is a fair amount of consistency in the responses
of different people in pitch perception experiments with simple stimuli, the ability to
sketch schematized pitch contours of aurally presented speech is very much a matter of
training and practice; novices at it often make very gross errors (when their estimates
are compared to an f0 contour). There are actually two parts to the assumption
that the perceived pitch of simple signals is straightforwardly generalizable: first, that
little memory is involved in the pitch percept, so that when analyzing a dynamic
signal we can excise short portions and pretend they are steady-state; and second,
that the additional fact that the dynamic signal happens to be speech doesn’t affect

26When I say this, I do not mean that people are actually wrong; people are infallible in their truthful
testimonies to their mental states. For them to be wrong would require the existence of another
independent definition of pitch. What I mean is that they are inconsistent, thus indicating uncertainty;
they might say that two tones of different frequency (mathematical pitch) have the same pitch, or they
might respond to the same stimulus differently upon repetition of an experiment, or they might give
the same response to different stimuli.
the computation. Whatever one's opinions on these questions, one must admit that the answers are far from certain, so that the output of any psychoacoustic pitch tracker on speech signals must be held in extreme distrust, at best.

Even if we knew what psychoacoustic pitch was and we could make a perfect tracker for it, there would still be many applications where it would be unsuitable. I think that there is a prevalent (usually unconscious) feeling that psychoacoustic pitch is really what pitch trackers should be getting, if only they could. But neither the fact that psychoacoustic pitch is hard to obtain, nor the fact that it has some claim to being the "correct" interpretation of the word "pitch," means that it is necessarily better for speech work. For example, many people live out their lives never realizing that there is a voiced/voiceless distinction in their language, or that there is such a thing as "intrinsic pitch"—they never hear it. Thus, at least at the conscious level of pitch perception (the only level we have data for), it would seem doubtful that psychoacoustic pitch could help much in studying voicing or intrinsic pitch. Indeed, one of the primary interests in a psychoacoustic pitch tracker is that it does throw away information; pitch tracks which bore a simple deterministic relation to f0 would be of no interest or benefit. But that means that an implementation of even a full theory of psychoacoustic pitch will be of little help in anything dependent on the source rather than the destination of the signal: spectral estimation, the study of micromelody, etc.

The applications of psychoacoustic pitch depend on the bifurcation mentioned above between production and reception. In most of the applications of production pitch this bifurcation does not apply; they cannot be done with psychoacoustic pitch because they depend on the inherent asymmetry between production and reception brought on by the direction of time. The applications that might benefit from a psychoacoustic pitch tracker, if there were such a thing, are:

- Linguistics: studying intonation.
- Speech recognition: using intonation.
- Language instruction and speech therapy.

1.4.3 Flaws

The distinction between theory and implementation, and the distinction between the two main reasons to make an implementation, are vital to any consideration of what a flaw in an implementation is, and how that flaw should be interpreted.

First of all, let us eliminate from discussion spurious flaws which are founded, for example, on misunderstanding of the input constraints on a pitch tracker. At best, these do not question the pitch tracker, but the chosen definition of pitch.

Valid flaws can be split according to the distinction made earlier between implementations motivated by theory and those motivated by application. What an engineer would regard as a failing in a pitch tracker might not be of any interest theoretically; a "flaw" for him could correspond to the failure to meet any of the requirements on space,
cost, weight, etc. The speed with which an implementation runs, for example, has no bearing whatsoever on one's judgment of the validity of the model. A pitch tracker which took two days to process a single utterance, for example, might be considered by some people to be much inferior to one which took only one minute, even if the second one might not be quite as accurate. From the point of view of a theory, however, the first pitch tracker could conceivably not be flawed at all, if it consistently came up with the right answer when it finally finished executing.

When it is determined that the performance of an implementation is flawed in a theoretically significant way (i.e., the actions of the implementation have discrepancies with the data), it is important to keep in mind the difference between theories and implementations, as outlined in Section 1.3.2. The implementation could be responsible for the theoretical failure via one of the distinguishing factors above: there could be a bug (1), an unreasonable approximation could have been chosen (3), or an inappropriate guess to complete the theory could have been made (4). Or the theory itself could be wrong, or both the theory and implementation could be responsible. A theory could be perfectly correct, and still have an implementation which has discrepancies with the data, because of the approximations necessary to conform to the artificial and irrelevant constraints imposed by the implementation medium. Also, one model could be more correct than another, while having an implementation that does not perform as well as that of the other.

1.4.4 Testing

There has been an incredible variety of methods used to evaluate the accuracy of pitch trackers. They have been evaluated by comparing them to a standard: to manually determined pitch contours (from either markers in the waveform, or from tracing a narrow-band spectrogram), to contours from slow but supposedly good pitch detectors (usually based on the cepstrum), or to contours from interactive semi-automatic methods ([90]). They have also been evaluated by scoring: visual inspection of the produced pitch contours, or listening tests of a vocoder driven by excitations derived from the contours ([62]). Yet another method has been to test pitch trackers on synthetic speech, and to compare the produced pitch contours with the contour corresponding to the original synthesizer excitation.

In light of the discussion in Section 1.3.2, from a scientific (as opposed to application oriented) point of view, there is no question how testing should be done; as with any implementation of a theory, the implementation should be tested against the sort of data that the theory is attempting to explain, and from which it was constructed. Thus pitch detectors of production pitch should be tested against data on production pitch (as determined from an EGG signal, for example); pitch detectors of mathematical pitch should be tested against data on mathematical pitch (as derived from the definition of quasi-periodicity); and pitch detectors of psychoacoustic pitch should be tested against data on psychoacoustic pitch (pitch perception experiments).
It is pointless to compare pitch trackers based on different theories, because they are attempting to model different data, so that of course they will give different results. In attempting to find a universal standard for pitch trackers, and in testing pitch trackers against something other than their own data, we are in effect comparing apples and oranges, and doing it by checking how much the fruits' juice tastes like lemonade.

It is important to keep in mind that when trackers are tested, for example, by comparing their outputs with manually produced pitch contours, surely it is not being assumed that the pitch tracker or definition of pitch should incorporate obscurities about the human visual system, or mistakes and misunderstandings of the human operator. As long as a distinction is maintained between the quantity that the human is trying to find, which is what the actual standard for the tracker is, and what he does find, there should be no problem. If this distinction is lost, then a new definition of pitch is being introduced: "the pitch of a speech signal is what a human examining the waveform (or spectrogram, or whatever) tells you it is."

From an application oriented point of view, it is not as clear how trackers should be tested. As was suggested in Section 1.3.3 with regard to "vocoder pitch," it is dangerous to rely entirely on tests dependent on how the tracker is being used; these tests should be moderated with other tests which are based on the underlying definition of pitch. It is possible to go far in answering the question "How good is this pitch tracker?" without having to ask "Good enough for what?"

Choosing the correct standard in testing is the most important (and one of the most neglected) aspects of testing pitch trackers. There are, however, a few other issues which should be brought up. First of all, some appropriate way of classifying the discrepancies from the data must be found. It is necessary here to distinguish as much as possible between errors brought on by unusual or difficult circumstances in the signal, and errors caused by problems with the implementation. This kind of confusion, for example, often comes up in discussions of "pitch-doubling" errors; it often isn't apparent whether a phonatory phenomenon is being talked about, or an irrelevant and unnecessary fact about the pitch tracker (caused by one of the problems outlined in Section 1.1.2.1). It is also important to be careful not to use error measures which are too sensitive to the particular postprocessing used. For example, it is not hard to convert a pitch tracker which is 100% accurate 90% of the time to one which is 90% accurate almost 100% of the time. Thus the distinction between "fine" and "gross" errors is not always very informative, and there is a great danger of effectively only comparing different post-processing algorithms (which are often interchangeable) rather than the basic pitch tracking algorithms. One should also be open to the possibility that error types, like standards, may well change from one pitch definition to another. It is meaningless to speak of "voiced-to-unvoiced" and "unvoiced-to-voiced" errors for mathematical and psychoacoustic pitch trackers. 27

Another issue lies in how error data should be summarized. There is a strong

---

27As we saw in Section 1.1.1.1, it is also close to meaningless to speak of these errors for production pitch trackers, but that is a different matter.
tendency in the literature on both pitch detection and speech recognition to reduce everything to a single error rate. There are several reasons that this is often not a good idea. First of all, these numbers are usually dependent on the particular data used, and are often not very useful in predicting how the pitch tracker will work if the speaker, speaking rate, phonetic balance, $f_0$ variability, etc. were changed much. Secondly, from a purely statistical point of view these numbers are often useless. The occurrence of an error from a pitch tracker at a particular moment does not represent a poisson process; the middle of steady state voiced sonorants is easy to handle, and if voiceless parts are ruled out as irrelevant as well, then not much of the test data even gives the pitch tracker an opportunity to fail. Thus to make a significant estimate of the error rate, a sizable amount of data is necessary. Thirdly, by collapsing everything into a single number, a lot of valuable information about the pitch tracker is lost. The idea of an error rate really only makes sense in the context in which it was originally formulated: detection of a pattern in noise, or decision making from random variables. It is unsuitable when there is a known systematicity in the testing data. If the pitch tracker always fails on nasals, it is much more useful to indicate that fact, then to say that it fails 5% of the time.

It isn't clear even how error rate should be defined. If fine errors are included, then all pitch trackers are always wrong; so some kind of threshold is necessary. Also, a decision will have to be made as to whether the rate is per second, per number of voiced frames, per pitch pulse, etc.

All of these problems with testing are difficult, and they cannot be eliminated entirely, but a great deal can be accomplished by following the recommendations that (1) lots of data should be obtained, with several speakers (of different ages and sex), spoken with variable $f_0$ and speaking style, using text carefully chosen for its phonetic content, and (2) error measures and error summaries should be relative to the correct standard, and should keep distinct errors which reflect properties of the pitch tracker, of the postprocessing, and of the data.
1.5 Previous Work

In this section, I will review the previous work on the topics discussed in this chapter. I am not concerned here with the large amount of work done on developing pitch trackers, but with the comparatively small amount of work done on the preliminaries to that effort.

First I would like to discuss a model often used in speech communication, commonly referred to as the “speech chain” ([25]). The idea is not attributable to a single person; it pervades most writing on speech communication. The three different kinds of pitch described in Section 1.1 each appear in some way at some point in the process of human verbal communication; the speech chain is a simplified view of that process, and it thus provides an alternative viewpoint to the typology given earlier. Because I think that in a discussion of pitch this viewpoint adds more confusion than it dispels, I have so far avoided its use.

The “speech chain” refers to the notion, familiar from classical communication and information theory, that messages are sent between humans from a transmitter, across a channel, to a receiver. The “transmitter” formulates his thought, which is converted into a string of lexical items, which in turn determines certain physiological control signals which manipulate the articulators and the glottal source, causing an acoustic compression-rarefaction wave to emanate from his mouth. This acoustic signal is then carried over the air—the “channel”—to the ear of the “receiver.” It is then mechanically transduced into a wave in the liquid-filled cochlea, which then tonotopically excites the hair cells, which after further processing finally causes the thought to appear in the receiver’s brain. Thus we now see the connection: the three stages in the “speech chain”—transmitter, channel, and receiver—correspond roughly to instantiations of the three main kinds of pitch that I presented earlier—production, mathematical, and reception.

The “speech chain” is useful for expository purposes, and I am certainly not about to suggest its proscription. But this simplistic view might easily lead to misconceptions about speech and pitch.

It gives a woefully inadequate, if not down right misguided, picture of language and thought.

The analogy with communication and information theory can easily be carried too far; to use any of the results from those fields it is necessary to make assumptions that are usually inappropriate and often damaging.

In thinking of this as a “chain,” which is symmetrical, it is important not to lose sight of the role that causality plays: Bill’s hitting John is not explained by the fact that John later had a bruise. The signal we have immediate access to is obtained by intercepting the message as it is going across the channel; as discussed in Section 1.4.2.3, the mouth and the ear do not have the same connection to the signal there. The production system alone is responsible for the signal’s appearance; the only connection the reception system has to it is a teleological one, as a possible source of constraints on the signal’s form.
The idea of "encoding" or "decoding" stages could be misinterpreted as an oversimplification of the process. As we saw in the discussion of voicing, the relationship between successive stages is far more complex than a simple transformation of medium. Similarly, it is not obvious that the signals at the two ends of the chain are identical, or even that their structures are the same.

For pitch, I believe that the observation that the different kinds of pitch are related in this way obscures the important distinctions between them. Another serious problem with the "speech chain" is that it is a chain, and a closed system. This linearity defeats the effort to make the input and output constraints clear. It also limits the generality of the whole discussion of pitch; while it is true that in this thesis I concern myself mostly with acoustic signals originating from human speech production, the classification of definitions of pitch includes more than just the signals occurring in the "speech chain."

The literature in this area is vast, but virtually all of it concentrates narrowly on one of physiology, pathology, psychophysics, linguistics, or engineering. To my knowledge, there have been only a few works with goals somewhat similar to those of this chapter. Chronologically first is Norris McKinney's 1965 research report for the University of Michigan Communications Science Laboratory, entitled "Laryngeal Frequency Analysis for Linguistic Research" [63]. On the abstract side, the only definitions he gives are of "laryngeal pulse frequency" and of "laryngeal fundamental frequency," neither of which is retained in this thesis. "Laryngeal pulse frequency" is the 1/t0 definition that I criticize in Section 1.2.2. "Laryngeal fundamental frequency" isn't very clear, but it seems to be an attempt at describing what I call mathematical pitch, in terms of laryngeal pulse frequency. On the practical side, the technological resources at the time were so limited that only a few of the algorithms or devices described are of present interest.

A more recent contribution is Sreenivas's 1981 Ph.D. thesis at the Indian Institute of Technology, "Pitch estimation of aperiodic and noisy speech signals" [113]. The stated motivation for this work was to consider both the psycho-physical and the "signal processing" approaches to the problem of pitch determination, hoping that by doing so he could solve some of the basic problems of pitch (p.I-2). This means that throughout the thesis he alternates between a discussion of the signal processing problems in determining mathematical pitch, and a discussion of psychoacoustic pitch perception facts. He regards this confounding of the two kinds of pitch to be a positive feature (p.I-2, p.I-32, p.III-1, p.V-1). As he has to, he plays down the importance of the distinctions between perceived pitch and the quasi-fundamental. For example, in his perceptual experiments he studies only quasi-fundamental match error rates, and not shift effects; he does not consider the subsidiary pitch percepts; and he uses only stimuli already known to have a residue related to the quasi-fundamental (speech-like, with many harmonics, and with quasi-fundamentals in a restricted range). This fusing of

---

28 There is extensive untranslated work by Baronin in Russian. However, Hess includes substantial excerpts from Baronin's papers, so it is possible to get a fair idea of what he has done from [40].
the two concepts has other consequences as well; he appears to connect inharmonicity in spectral peaks with JND and masking effects in perception, and he similarly relates an inability (visually or with simple signal processing) to find a spectral peak with the inability of the ear to find it as well. He also brings in production pitch by speaking of "successive excitations" and by using the source of an LPC synthesizer as his idea of the correct quasi-fundamental. He seems to equate voicelessness, non-quasi-periodicity, and a lack of a pitch percept (p.V-3). As I described in section 1.3.3, neither mathematical pitch nor (at least now) psychoacoustic pitch should be considered as a phenomenon special to speech signals. The fact that he does restrict his discussion to speech severely limits the scope of his notion of quasi-periodicity; the only way he thinks of aperiodicity is in terms of a speech production model (p.II-4).

Fortunately, he uses psychoacoustic pitch facts so loosely that when his work is stripped of the influences of psychoacoustic pitch, there is still a good algorithm left for determination of mathematical pitch. Also fortunate is that his algorithm does not make any significant assumptions special to speech signals (though that is the only kind he ever considers), so it is also free from the influence of production pitch. This algorithm will be considered in Section 2.2.

In what he calls a "functional demarcation of pitch" ([112], [113] p.VI-1ff.), Sreenivas identifies three different characterizations of pitch, and suggests that both pitch definitions and pitch tracker evaluations should be treated differently for each of the three cases. So far this sounds similar to what is being proposed here, but his partition is much different from mine. He separates his definitions according to the assumptions in the signal processing used in implementations:

1. Speech is in general quasi-stationary; hence, voiced speech can be assumed to be stationary and periodic over short durations of the order of 30-40 msec; pitch corresponds to this periodicity and can be estimated by an appropriate transformation of the signal, such as used for perfectly periodic signals. [correlation, cepstrum, maximum likelihood methods]

2. The short-time spectrum of voiced speech exhibits harmonic structure (similar to that of a periodic signal); pitch corresponds to the fundamental frequency of this harmonic structure and can be deduced from a knowledge of the harmonics. [methods based on processing spectral peaks, such as his own]

3. The waveform of voiced speech exhibits "peakedness" and "regularity" from one period to the next, corresponding to the pitch-epochs (instants of major vocal-tract excitation); such epochs can be identified by pattern recognition techniques similar to those used for visual identification. ["time-domain" techniques] ([113] p.VI-2)

Although this is a step in the right direction, it is not that much different from any other partition of pitch trackers according to their algorithmic properties.

Like McKinney's report, Hess's book is first and foremost a literature survey—"encyclopedia" might be more appropriate in this case. I think it would be fair to say that Hess's book is the second edition of McKinney's; it brings it up to date. As a research aid for the subject it is a godsend; his bibliography is particularly helpful. Moreover, through his summaries he makes available the content of many papers whose original would be time-consuming to get access to: foreign literature, unpublished dissertations, laboratory memoranda, patents, etc. He also provides some interesting historical notes, good introductions to signal processing and physiology, and a discussion of computational considerations and of application areas.

Consistent with his goal of describing the state of the art, for reasons of exposition Hess also articulates the often unspoken assumptions of past researchers. But unfortunately he absorbs these as uncritically as he does the algorithms themselves, and this is reflected by confusions throughout the book. His analysis is vitiated by the fact that he follows the common practice of organizing his discussion by algorithmic properties. He does not develop the important consequences that the pitch definitions have on pitch tracker design, testing, applications, etc. He does not distinguish between definitions of pitch which are of objective interest, and definitions which arise only from the limitations of an implementation. For him, definitions of $t_0$ which average over several periods, and definitions of $t_0$ which are relative to an arbitrary point in the glottal cycle, are of equal status with the other definitions, apparently because there are pitch trackers which do things that way (pp.476-477). He also admits notions of $t_0$ dependent on structural waveform primitives in the speech signal. With this point of view, it is hard to see how he avoids the fallacy of defining pitch in terms of what a particular tracker returns. Moreover, by conflating the definitions of pitch with particular implementations, problems with the implementations translate for him into faults of the definition. Although he partially recognizes in his last chapter the importance of the different kinds of pitch (pp.472-477), he regards these as just different views of the same problem; he still would like a "general solution" (p.547-549). Because he considers all pitch trackers to be trying to solve the same problem, though in different ways, he is lead to criticize some definitions/implementations because they can't handle the same inputs that others can, and he suggests that pitch trackers based on different definitions be used together so as to complement each other when one of them fails (p.548). When he attempts to elucidate the different kinds of pitch, he is thwarted by his idea that they are all basically the same thing:

For speech signals we can widely assume that virtual pitch and fundamental frequency correspond to each other. This is valid as long as the fundamental frequency $F_0$ is understood as the reciprocal of the fundamental period $T_0$. (p.76)

In this book the definition (2.34) of fundamental frequency as the reciprocal of fundamental period $T_0$ will always be obeyed....That does not mean, however, that the concept of fundamental frequency as the largest common divisor of the partials is invalid. On the contrary, the pitch perception theories apply the principle of subharmonic matching—which actually
represents the search for a common (perceptual) divisor for the individual spectral pitches—for determination of perceived pitch. (p.78)

Thus in the space of a few sentences he manages to twice confound all three types of pitch. He believes that voicing determination is a separate problem from pitch determination, and is preliminary to it (p.34), a position that I argue against in Section 1.1.1.1.

I should make it clear after this vituperation that although these problems are quite serious, they apply to an aspect of Hess’s book which is peripheral both to its goal and to its content. It remains an outstanding and scholarly presentation of virtually every aspect of the past work on pitch determination, and these defects on an abstract level detract only slightly from it. The reason I criticize him in particular for these parts is not that he is worse than others in this area, but that he is one of the few people who says anything explicit about it.
1.6 Summary

The approach I have taken to the problems of pitch determination is to first study carefully what pitch and pitch trackers are, and then expose the consequences this has for further questions. Picking up on the acknowledged existence of ideas of pitch other than psychoacoustic pitch, I have provided a complete typology of pitch definitions, partitioning them into the categories of production pitch, mathematical pitch, and reception pitch. I have tried to clarify the connection between voicing and production pitch. I have given a definition of quasi-periodicity, and I have assessed the alternative definitions underlying the bulk of short-term analysis pitch determination schemes. I have proposed an explication of f0, in terms of the general problem of converting a repeating sequence of discrete events into a time-varying rate function. I have intimated that for speech, currently there are only two reasonable goals for an implementation in the field of pitch determination: recover the production pitch of the signal, represented as a listing of when glottal closures occur, or pretend that the signal is quasi-periodic, and determine the quasi-fundamental. I have developed some of the consequences the typology of pitch definitions has on what constitutes a suitable input to a pitch tracker, on the representation of the different kinds of pitch, and on the applications of pitch trackers. I have shown the importance of the distinction between theory and implementation, and between theoretical and application reasons for constructing an implementation. I have indicated some of the consequences these distinctions have on the interpretation of flaws in an implementation, and on how evaluation should be done. Finally, I have discussed and criticized past attempts to resolve some of these issues.
Chapter 2

Pitch Determination

The goal of this chapter is to propose solutions to two of the implementation problems extant from the previous chapter: determining the glottal closure moments from a speech signal, and determining the quasi-fundamental of a quasi-periodic signal. Some of the current and potential applications of the (partial) solutions to these problems are discussed in Section 1.4.2, so I will not repeat them here.

2.1 A Production Pitch Tracker

2.1.1 Background

Of the pitch trackers developed in the past, it is the so-called “time domain” class of pitch tracking algorithms which is most relevant to the project of trying to find glottal closure locations from the speech wave. It is not too much of an exaggeration to say that so far these have fallen into two categories: (1) those which are fairly reliable pitch trackers, but which do not find the glottal closure locations; instead they find the nearest waveform zero-crossing or maxima, and (2) those which do not really constitute pitch determination algorithms, but are just transformations of the speech signal which give reasonably salient peaks at glottal closures for steady state voiced sonorant speech; almost always the method is not carried through to actually find the locations automatically from the processed signal.

The “trackers but no closures” category includes some of the earliest pitch detection work, done with analog technology, and described thoroughly by McKinney ([63]). These proceed by first trying to isolate the fundamental either by low-pass filtering or by nonlinear enhancement, and then determining the periods by finding where the signal crosses some threshold (often zero). By all accounts, these were not very successful. They were largely supplanted by the techniques which were made possible with
digital speech processing. The algorithm by Reddy ([92]) set the tone for all future pitch trackers of this category. First, maxima and minima in the waveform were found which satisfied certain simple criteria; Reddy tested the relationship between successive amplitudes and he imposed a minimum period limit. "Significant peaks" were near cooccurrences of accepted maxima and minima; the sequence of significant peaks was taken to be the tentative answer. This list was then corrected by moving along it left to right considering successive markers: the relative error between successive periods was calculated, and according to the value of this error the marker was accepted, replaced by one nearby, rejected, or another marker was inserted before it. Miller's algorithm ([64]) is similar, except it uses sequences of nonzero values ("excursion cycles") as the primitives, and it accepts them according to their area. The well-known algorithm of Gold and Rabiner ([38]) has the same basic structure, but it uses six different primitives, and it determines the period by a complicated voting procedure on the intervals between like primitives. The algorithm of Tucker and Bates ([120]) is an extension of the Miller algorithm; for each peak which survives adoptive center clipping, five features are computed—amplitude, width, energy, polarity, and aspect ratio—and on the basis of those, context constraints are applied to remove all but the peaks from the pitch period beginnings. Finally, Philips ([83]) takes this progression to its logical conclusion, by first measuring many different features of correct peaks from training data, then examining the resulting multi-dimensional feature space to try to find the optimal binary decision criterion for accepting or rejecting a peak, and then using this as the basis of a pitch detector.

A problem with all of these, whether or not they reliably determine rough estimates of period length, is that the primitives that are being used are not really indicators of the beginnings of pitch periods. As was discussed in the previous chapter, we have a choice: we can design an algorithm which will work for speech, music, and other quasi-periodic sounds, in which case there is no universal notion of a beginning of a period, or, if we want to find period beginnings, we must restrict ourselves to speech, and define the beginning in terms of how speech is produced. These algorithms, while coming close to the second of these choices, still do not go so far as to appeal to the underlying phenomena in the production of the signal, but instead use surface structural properties of the speech wave. This introduces an unnecessary amount of spurious noise in the resulting period estimates, because glottal closures are not timed systematically relative to the biggest peaks or the sharpest zero crossings.

The "closures but no trackers" category consists of transformations of a speech signal which attempt, by inverse filtering or similar means, to reconstruct the acoustic excitation at the laryngeal source which corresponds to that speech signal. Because this is so difficult, often the goal is simplified somewhat, so that instead the task is just to determine a signal which reflects certain important characteristics of the true excitation—in particular, one which reflects the moments of glottal closure. Many of the papers below go on to consider methods of inverse filtering, using the transformed signal to locate the analysis windows (semi-automatically); however, for our purposes we are only interested in the transformations. As explained in Section 1.4.1, necessarily
these methods assume properties of the input signal which are special to (idealized) speech signals. Ananthapadmanabha and Yegnaarayana ([6]) use “epoch extraction,” a notion taken from signal detection theory ([133]), to compute a signal which will hopefully have large excursions at the moments of epochs/glottal closures. The epochs of a signal are defined as the points of discontinuity of the lowest order derivative of the signal which has discontinuities. According to the theory, the optimal epoch filter which emphasizes these discontinuities basically amounts to the Hilbert envelope of the output of a bandpass filter with a high center frequency. In their paper, they use one-third octave bandpass filters centered around 10 and 20 kHz. Both the theory and the implementation are somewhat complicated, and it would appear to be quite sensitive to noise. The connection between epochs and glottal closures is based only on the fact that glottal closures are associated with discontinuities. Perhaps for these reasons, the same authors later came out in favor of using linear prediction coding (LPC) error residual ([7]). They tried to remove some of the ambiguity in the residual by putting it through a bandpass filter (a Hanning window covering the whole frequency range below Nyquist) and then calculating its Hilbert envelope. Atal and Ifanaier ([9]) put the prediction residual through a 1000 Hz cutoff lowpass filter. In this case, they actually did make a pitch detector using their transformation; they used a simple peak-picking method and a voiced-unvoiced decision based on the ratio between the rms value of the speech and the rms value of the residual. They did not investigate the capabilities of this algorithm very extensively, however. Strube ([116]) considered a variety of transformations based on LPC: (1) one method just uses the unaltered LPC residual; (2) in another, LPC optimization is done for successive very short regions (say 3 msec), and the ratio between the total mean-squared prediction error and the mean-squared signal is calculated; regions which contain a glottal closure should have a high value for that ratio; (3) a third method, due to Sobakin ([108]), is to use the logarithm of the determinant of the autocovariance matrix calculated over a short region; the determinant should be small when prediction is good and therefore linear dependence is high. He also tried several modifications of Sobakin’s technique to reduce its dependence on signal amplitude; his main conclusion was that the idea was both theoretically and empirically questionable. Wong, Markel, and Gray ([131]) use a transformation similar to Strube’s second one. Although it appears from the discussion in their paper that the procedure might be automatic, if it is they do not test its properties by objective standards, nor does it seem that they have developed it for use on sounds other than steady state vowels. Sondhi ([109]) suggested putting the speech signal through a bank of 32 wideband filters; their outputs were individually normalized by their short-term rms values, phase-compensated to remove a minimum-phase component, and then added together again. The motivation for this was to remove the amplitude and phase effects of the vocal tract, to recover the impulse-like excitation. This is similar in spirit, but much different in method, to Oppenheim and Schafer’s proposal ([77]) of homomorphic (cepstral) filtering to remove the transfer function of the vocal tract. Finally, DeMori et al. ([23]) use syntactic pattern recognition on the rms outputs of a filter bank to identify the glottal closures.
(they looked for simultaneous peaks). The published description of the performance of the algorithm is mostly anecdotal, so it isn’t clear how effective their rather exotic approach is.

Given this historical synopsis, it is clear that what remains to be done is to combine the positive aspects of both of these traditions; we would like to use a technique which is capable, in principle at least, of accurately determining glottal closure locations, but we also want to demonstrate that the transformation gives a signal which is more than visually perspicuous—that it is possible to find the locations automatically as part of a general-purpose pitch determination algorithm. In the following pages, an attempt to reach this goal will be described. Several, though by no means all, of the ideas and techniques described below are not new; they have been culled from the voluminous literature on the topic of pitch determination. The problem has been worked on for a long time, and it would be surprising (though certainly not inconceivable) that a basic signal processing method had somehow been overlooked, and had not been used sometime, somewhere before. Rather than pursue gratuitous novelty, I have instead concentrated mostly on intelligently choosing from among past ideas, and on combining them appropriately to produce a coherent algorithm.

### 2.1.2 The Stages of Processing

In the approach that I have taken, the input speech signal is processed in a succession of distinct stages. In the first stage, the “basic transformation” is applied; the general goal of this transformation is to produce a signal which should have large excursions at glottal closures and less activity elsewhere. I consider three possibilities for this: computation of the LPC residual, computation of the normalized total prediction error for a short region, and cepstral inverse filtering. Typically these signals are noisy, have multiple peaks for a single closure, and have peaks of both polarities. It is therefore necessary to simplify this signal and reduce the number of potential peaks which need to be examined. Three methods of doing this are considered: lowpass filtering, computing short-term rms values, and computing the Hilbert envelope. In the third stage, an algorithm is used on the simplified signal to locate (hopefully) all and only the peaks corresponding to glottal closures. The simplification done in the second stage slightly shifts the locations of the corresponding peaks in the unsimplified signal. It therefore might be necessary after the second stage to adjust the locations found, by replacing them with the corresponding maxima locations before simplification. A question which will be addressed in Section 2.1.2.4 is whether this step improves or worsens accuracy. Finally, an effort is made to deal with the areas of the speech signal where the basic transformation failed, by growing out from the regions which have been successfully labeled. An example of this processing is shown in Figure 2.1.

These stages are relatively independent; the choices which must be made for each of them can mostly be made on the basis of criteria which do not involve the choices made in the other stages. In the next few subsections, I discuss each of them individually.
Figure 2.1: Some of the stages of the processing used in the production pitch tracker are shown above. The input is a digitized speech signal; here a male vowel /a/ is shown on top. The basic transformation, zero-phase bandpass filtered LPC prediction residual, is shown in the second graph. In the next stage, the output of the basic transformation is simplified to facilitate peak picking; the method shown here is to square the signal, smooth it, and then locate all the maxima. In the third stage, shown in the bottom graph, the spurious maxima from the previous stage are eliminated. The next stages, which are not represented here, are (optionally) adjustment of peak locations, and extrapolation from stable regions.
2.1.2.1 The Basic Transformation

As described in the previous section, there are several transformations which have been suggested as a means for computing from a speech signal another signal which has marked activity centered closely around moments of glottal closure. Three of these have been selected for investigation.

The first two of these are based on LPC. There are excellent introductions to linear prediction elsewhere (e.g. [91], [58], [131]), so I do not propose to describe it here in a way which will be understandable to those not already somewhat familiar with the topic. The problem in linear prediction is to come up with a set of fixed coefficients \( a_k \), say \( p \) of them, which for a short (steady state) region of the speech signal can be used to predict samples:

\[
e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \iff E(z) = S(z) \left(1 - \sum_{k=1}^{p} a_k z^{-k}\right) = S(z)A(z) \tag{2.1}
\]

Here \( s(n) \) is the sequence of speech samples, and \( e(n) \) is the prediction error for the \( n \)th sample of \( s \). When the criterion for the evaluation of a particular set of \( a_k \) is to minimize the total prediction error \( E_{\text{total}} = \sum e^2(n) \) over some range of \( n \), this turns into a straightforward linear least-squares optimization problem: \( \Phi a = \Psi \), where \( \Phi \) is the autocovariance matrix of the relevant subsequence of \( s(n) \), \( a \) is the vector of the \( a_k \), and \( \Psi \) is the autocovariance sequence. If \( s(n) \) is windowed around the region of interest, then \( \Phi \) is the autocorrelation (Toeplitz) matrix and \( \Psi \) is the autocorrelation sequence, giving slightly different results. There are a variety of efficient techniques for solving this system of equations to get \( a \), which take advantage of the special properties of \( \Phi \). Now a voiced sonorant speech signal can be modeled as the following cascade of linear systems, which for a short-term computation can be assumed to be time-invariant:

\[
s(n) = p(n) * g(n) * h(n) * r(n) \iff S(z) = P(z)G(z)H(z)R(z). \tag{2.2}
\]

\( p(n) \) is an impulse train, and \( g(n) \) is the shape of a single glottal pulse, so that \( p(n) * g(n) \) is the glottal volume velocity. \( h(n) \) is the impulse response of the vocal tract, making \( p(n) * g(n) * h(n) \) the volume velocity at the lips, and \( r(n) \) is the radiation characteristic giving \( s(n) \), the pressure wave measured 30 cm or so from the mouth. \( G(z) \) can be modeled by an FIR (only zeros) or a 2-pole system, for non-nasal sounds \( H(z) \) can be assumed to be an all-pole system (one pole pair per formant), and \( R(z) \) can be modeled as a differencing filter \( 1 - z^{-1} \). Therefore, except for the single pole at 0 Hz in \( R(z) \), and the broad poles of \( G(z) \), all the poles in \( S(z) \) are due to \( H(z) \). Thus we might hope that if we perform the optimization to solve equation 2.1 for the \( a_k \), the \( p \) zeros of \( A(z) \) might be exactly all the poles of \( H(z) \), so that:

\[
H(z) = \frac{1}{A(z)} \tag{2.3}
\]
and

\[ P(z)G(z) = \frac{E(z)}{R(z)}. \quad (2.4) \]

Thus we might hope to obtain the glottal volume velocity \( p(n) \ast g(n) \) of a steady-state voiced non-nasal sonorant by solving for the predictor coefficients \( a_k \), applying the inverse filter \( A(z) \) to \( s(n) \) to get \( e(n) \), and running \( e(n) \) through an integrator \((1/(1 - \mu z^{-1}) \) with \( \mu \approx .99 \) is one approximation to an ideal integrator and to \( 1/R(z) \)).

That's the theory anyway. We would expect glottal closures to cause discontinuities in the slope of the glottal acoustic flow, because when closures occur the glottal folds are abruptly stopped in their motion towards each other. Thus we would be interested in the second derivative of \( p(n) \ast g(n) \). Because \( e(n) \) is already approximately equal to the first derivative of \( p(n) \ast g(n) \), this would suggest that we should consider using the result of differentiating \( e(n) \) once. However, it turns out that this extra differentiation step is unnecessary, for the following reason. In the solution of the LPC equations, an attempt is made to find the coefficients \( a_k \) of the all-pole spectrum \( H(z) \) on the basis of \( S(z) = P(z)G(z)H(z)R(z) \). As we have been saying, \( R(z) \) introduces a +6 dB/octave tilt to the spectrum. The glottal source spectrum \( G(z) \), because of the second order discontinuity, falls off at roughly -12 dB/octave. \( P(z) \) has no effect on the tilt of the spectrum. Thus the net effect is that \( S(z) \) has a -6 dB/octave rolloff which \( H(z) \) does not have. This can have a pernicious effect on the ability of the optimization procedure to correctly determine the higher frequency poles of \( H(z) \), at least for correlation LPC. For this reason, it is standard practice to preemphasize the speech signal \( s(n) \), usually with a differencing filter \( 1 - \mu z^{-1} \). This improves the quality of the estimate of the parameters of \( H(z) \). If the inverse filter \( A(z) \) is then convolved with the differenced signal \( s'(n) \), by linearity the net effect of the whole operation is to obtain an \( e(n) \) approximately equal to the first derivative of the sequence \( e(n) \) that would have been obtained without the preemphasis.

We now have one possibility for the result of the first stage of the pitch determination algorithm: the sequence \( e(n) \) derived by convolving \( s(n) \) with an FIR inverse filter, where the filter coefficients have been obtained by LPC optimization on the preemphasis-sized \( s(n) \). Because \( s(n) \) is time-varying, the inverse filter is recalculated for successive analysis frames; a 10 msec skip between frames is typical. The width of the analysis regions may be chosen independently of the distance between them; there is a tradeoff between the greater variability within longer windows, and the poorer efficiency of the estimate for shorter windows. Typical widths are in the 20-25 msec range.

Another possibility for the first stage, which is based on the same theory, is to calculate the optimal \( E_{total} \) for very short analysis regions (around 3 msec), starting at successive samples of \( s(n) \). One would expect the total prediction error to be greatest

---

1This isn't really true; LPC optimization always tries to flatten the spectrum, so that without preemphasis \( H(z) \) falls off at roughly the same rate as \( S(z) \) does. Thus \( e(n) \) without preemphasis is actually also roughly the second derivative of \( p(n) \ast g(n) \). At least, it is closer to that than to the first derivative.
for analysis regions centered around glottal closures, so that this sequence of error energies could be used as the output of the first stage. However, there is good reason to expect that this new sequence—call it \( E(n) \)—will not be as good as \( e(n) \). Consider what happens for a single analysis region of the "\( e(n) \)" method. Suppose that all the inverse filters \( A(z) \) calculated by the \( E(n) \) method for the very short subregions of a single \( e(n) \) analysis region, will be the same as the \( A(z) \) calculated by the \( e(n) \) method. In that case, it is clear that \( E(n) = \sum_{n=n_0}^{n_0+L} e^2(n) \), where \( L \) is the length of the subregions. In general, the sequence \( E(n) \) will behave much like the result of squaring the sequence \( e(n) \), and then smoothing it by convolution with a rectangular window of length \( L \). It would be preferable to have the signal \( e(n) \) directly, before it has been smoothed, so that the ultimate accuracy of the glottal closure location estimates will not be affected. In the second stage, when the transformed signal will be simplified, some smoothing is done, but this can be eventually counteracted by going back to \( e(n) \) if necessary; moreover, one is not constrained to just use a rectangular window of a particular width. At best they are equivalent; because of the extremely short analysis width, the second method will usually make poor estimates of \( H(z) \), so that the effective \( e(n) \) used will not be as good. It also takes longer to compute \( E(n) \) than \( e(n) \). For these reasons, \( E(n) \) was dismissed from further consideration for the first stage.

When LPC residual is used in practice several intrinsic drawbacks become apparent. First of all, a value for \( p \), the filter order, must be set. A common rule of thumb is that a pole-pair should be included for each 1000 Hz in the signal, with a few extra added for good measure. Thus a 12-pole LPC analysis might be use on speech sampled at 10 kHz (with a 5 kHz cutoff frequency). The danger of using too low an order is that the formants will not be canceled out. The danger of using too high an order is that some poles might be assigned to canceling harmonics, thus lowering the signal-to-noise ratio of \( e(n) \). For LPC synthesis, the second threat is less serious than the first, but in pitch extraction it is important that we have salient peaks in \( e(n) \), so one would want to use as low an order as possible. Because different speakers have different vocal tract lengths, and thus different average formant spacing, using a fixed value of \( p \) is problematic. Nevertheless, to avoid requiring this information from the user, or trying to figure it out from a few spectra of the input, a fixed value is in fact used. This can potentially cause problems for female speakers; they typically have shorter vocal tracts, so fewer poles are required in the analysis of their speech, and because they usually have higher \( f_0 \) values, there are fewer peaks in their speech spectra. Both of these factors make it more likely that a pole will be assigned to a harmonic. Despite these statements, the order of the LPC analysis is not a crucial parameter when the prediction residual is to be used only for pitch detection. As can be seen in Figure 2.2, the prediction residual looks about the same regardless of the number of poles \( p \); although the 14th order LPC residual looks terrible for high-pitched speech, so does the 6th order LPC residual. This observation is consistent with plots in the literature of normalized prediction error vs. order of predictor ([91], p.430).
Figure 2.2: The LPC residuals derived from using different numbers of poles are usually not significantly different. The upper set shows high-pitched female speech (the word "where") and the corresponding residuals; the lower set shows the performance on the sound /ŋ/.
Another much less serious problem is with the preemphasis: because the glottal source does not fall off at -12 dB/octave for low frequencies, because the radiation characteristic is only +6 dB/octave for the idealized omnidirectional point source in the far field, and because discrete implementations of differentiation are approximate, the spectrum will not be completely balanced.

LPC is often accused of consistently underestimating formant bandwidths; this of course has a deleterious effect on its inverse filtering capability. One might try to reduce the bad consequences of both this problem, and the problem of a pole being assigned to a harmonic, by modification of the coefficients $a_k$ after they have been obtained. One way would be to find the roots of the polynomial $A(z)$, and examine and correct the bandwidths individually. Another much faster method, which does not require rooting the polynomial, is to multiply each $a_k$ with $z_0^k$, $|z_0| < 1$. This has the effect of scaling down all the poles of $A(z)$ by $z_0$, which in turn increases their bandwidth; if we take $z = e^{i\omega T}$ (a crude approximation), let $T$ be the sampling period, and interpret bandwidth as the distance between the half-power points, then the new bandwidth of $s_p = j\omega_{aa} + \sigma_{aa}$ corresponding to the pole $z_0 z_p$ is

$$B = -\frac{1}{\pi} \sigma_{new} = -\frac{1}{\pi T} \log(|z_0 z_p|) = -\frac{1}{\pi} \sigma_{old} - \frac{1}{\pi T} \log(|z_0|). \quad (2.5)$$

Thus to increase the bandwidths uniformly by $W$ Hz, the coefficients should be multiplied by $e^{-\pi WT}$. However, as can (barely) be seen in Figure 2.4, the effect of this operation is to simply make prediction generally worse, so that $e(n)$ begins to look like a noisy version of $s(n)$; it does not differentially increase the size of the peaks at just glottal closures times.

The LPC model assumes an all-pole transfer function, which is incorrect for some voiced sounds: nasals and voiced fricatives. The LPC residual of nasals, or of any mostly low-frequency sound, such as a /u/, is typically not very good because those sounds can be predicted so well. The residual is quite small throughout these sounds, and sometimes the signal is not relatively greater at the glottal closure times than at the others. Some kind of bandwidth broadening (one more intricate than just multiplying the $a_k$ with $z_0^k$) might partially counteract this effect. Voiced fricatives result in a residual which is poor for pitch tracking; what regularity there is in the speech is often removed by the inverse filter, and the noise component, which can’t be as easily canceled, is predominant. This situation might be ameliorated somewhat by taking advantage of the fact that the noise in voiced fricatives does not usually cover the whole spectrum: typically /z/ cuts off below 4 kHz and /z/ cuts off below around 2 kHz; the much weaker (and therefore less problematic) fricatives /\d/ and /\v/ cutoff around 1 kHz. By just using the part of the spectrum where the glottal source is dominant, one might hope eliminate the contribution of the noise sources to the $e(n)$ sequence. In the usual implementation of LPC, all frequencies from 0

---

2This is not to say that LPC spectra of nasals are correct, it is just that with a dozen or more poles it is possible to fit the high energy parts of the spectra quite closely.
Figure 2.3: Shown here is part of the same nasal as Figure 2.2, and 14th order LPC residuals with different amounts of bandwidth broadening.
to $R/2$ Hz are modeled in the spectrum, where $R$ is the sample rate. The range could straightforwardly be changed to be 0 to $R/2k$ Hz, for some integer $k$, by lowpass filtering the input signal to remove frequency components above $R/2k$, and then doing a $k$th order decimation. The resulting signal, sampled at $R/k$, would then be used in the usual LPC processing. Another faster and more general technique is known as selective linear prediction ([56]). This is a method of computing the proper correlation coefficients to use in $\Phi$ and $\Psi$ to accomplish modeling of a frequency subrange. As is well known, the DFT of the (circular) correlation sequence is equal to $|S(k)|^2$, where $S(k) = \sum_{n=0}^{N} s(n)e^{-j2\pi nk/N}$ is the DFT of the input sequence $s(n)$ with length $N$. Thus one way to compute the correlation sequence is to take the IDFT of $|S(k)|^2$. One can instead take the IDFT of a subsequence of $S(k)$ to get a correlation sequence which corresponds to matching just those frequencies. For spectral estimation, this is much more flexible than the decimation method, because it can do bandpass as well as lowpass modeling, and because it can easily change the sampling rate by any rational factor expressible as the ratio of two feasible FFT orders. However, if it is desired to apply the inverse filter to $s(n)$ (in time by convolution), then this method is as constrained as the decimation method in its possible frequency ranges. Instead of lowpass or bandpass filtering $s(n)$ and generating an $e(n)$ from only modeling that sub-band, a much simpler alternative is to go ahead and do LPC on the whole band, and then filter the resulting $e(n)$. This second choice is inferior to the first only if the presence of the noise components in the high part of the spectrum significantly affects the quality of the modeling of the parts passed by the lowpass (or bandpass) filter.
Another problem with the fact that LPC models the whole range from 0 to \( R/2 \) Hz is that typically an anti-aliasing filter is applied to the speech before digitization, and this will affect the spectrum of the speech below the cutoff frequency. The comments which were made on removing the effect of frication apply to this problem as well.

The reason to remove high frequency components from \( e(n) \) is, as we have said, to try to eliminate some of the noise components. The LPC residual usually also has a low frequency baseline component; this could be due to \( g(n) \) (at best, we have \( \frac{d^2}{dt^2} p(t) * g(t) \)), or due to poor modeling of the low frequencies of \( s(n) \). Either way, it is desirable to eliminate it. Thus \( e(n) \) is put through a zero-phase bandpass filter; the result is shown in Figure 2.4. This is implemented by convolution, to avoid the time domain effects of a finite frequency-domain window and the complexities of frame by frame signal processing.

LPC (like cepstral inverse filtering, and most other methods) assumes that the transfer function does not change within the analysis region. In actual fact, when the glottal folds open the formants are usually dampened. A solution to this problem is to restrict analysis to the closed-glottis region—but that requires knowing where the glottal closures occur (catch 22). A possible use for this pitch tracker, as was pointed out in Section 1.4.2.1, is in improving spectral analyses like LPC.

Another problem with the LPC residual is the existence of "spurious" peaks in the signal. LPC is based on prediction of samples; therefore any unpredictable event in \( s(n) \) will show up strongly in \( e(n) \): not only friction and aspiration, but also bursts and releases of various sorts. There is also a general problem with the residual that because different speech sounds vary widely in how well they can be predicted, there is a big dynamic range in it; peak amplitudes which in one part of the signal might be noise, and much less than the amplitudes for closures, might be larger than the peak amplitudes for closures in other parts of the signal. Finally, LPC is not terribly flexible; one of the sacrifices which is necessary to get an efficient closed-form solution is that there is not much room for applying heuristics.

Another transformation to consider for the first stage is based on the cepstrum. In describing the method I will use the same summary style as for LPC; see [77], [91], and [124]. Here voiced sonorant speech is modeled as \( s(n) = p(n) * v(n) \), where \( v(n) \) is the combined response \( g(n) * h(n) * r(n) \). The goal is to obtain \( p(n) \) from \( s(n) \), and use that as the output of the first stage. The basic observation of cepstral techniques is that because \( \log[S(e^{j\omega})] = \log[P(e^{j\omega})V(e^{j\omega})] = \log[P(e^{j\omega})] + \log[V(e^{j\omega})] \), one might hope to separate \( \log[P(e^{j\omega})] \) from \( \log[S(e^{j\omega})] \) by linear techniques (this is called homomorphic signal processing). In particular, one could take the inverse DTFT of \( \log[S(e^{j\omega})] \), to get \( \hat{s}(n) \), the complex cepstrum of \( s(n) \). Because the DTFT is linear, \( \hat{s}(n) = \hat{p}(n) + \hat{v}(n) \). Because \( \log[V(e^{j\omega})] \) is slowly varying (as a function of \( \omega \) whereas \( \log[P(e^{j\omega})] \) is made up of impulses (for periodic \( p(n) \)), one might hope that their cepstra would therefore be separated: \( \hat{v}(n) \) would be a "low-time" signal, whereas \( \hat{p}(n) \) would be a "high-time" signal. If that were true, one could obtain \( \hat{p}(n) \), and hence \( p(n) \), by removing the low-time part of \( \hat{s}(n) \). Thus the essence of cepstral inverse filtering is to remove a transfer function found by smoothing the log spectrum of the signal. This is similar
to Sondhi’s method ([109]), except that he finds the transfer function by smoothing
the power spectrum; in both methods the phase of the spectrum which is divided out
is assumed to be minimum phase (stable all-pole systems have minimum phase).

Several important questions were skipped in the preceding short intuitive sketch.
First of all, \( S(e^{j\omega}) \) is in general a complex function, and the log of a complex
number is not unique. For \( \log |S(e^{j\omega})| \) to define a real, stable sequence \( \hat{s}(n) \), it turns out that in
the equation \( \log |S(e^{j\omega})| = \log[|S(e^{j\omega})| + j \arg[S(e^{j\omega})], \arg[S(e^{j\omega})] \) must be a continuous
odd function of \( \omega \). This means that phase-unwrapping is necessary, since the principal
arg of \( \log |S(e^{j\omega})| \) is not continuous; this is a notoriously difficult task. Fortunately
for many applications phase unwrapping is unnecessary, because the (real) cepstrum is
sufficient: \( c(n) = IDTFT[\log |S(e^{j\omega})|] \). \( c(n) \) is the even part of the complex cepstrum
\( \hat{s}(n) \), and it exhibits many of the same characteristics as \( \hat{s}(n) \). At first it would appear
that we can’t use \( c(n) \), since we want to obtain \( p(n) \) by finding \( \hat{p}(n) \), and we cannot
do that if we throw away the phase information of \( s(n) \). However, all is not lost, because we can take the DTFT of the high-time part of \( c(n) \), to get an approximation
of \( \log |V(e^{j\omega})| \). Next, we can take advantage of the fact that we know a priori what
the phase of \( V(e^{j\omega}) \) should be; because it is approximately the transfer function of the
vocal tract, it should be minimum-phase (its log magnitude and arg should be Hilbert
transform pairs). We can use this fact to construct \( V(e^{j\omega}) \) from \( \log |V(e^{j\omega})| \), and then
\( p(n) \) will be \( IDTFT[S(e^{j\omega})/V(e^{j\omega})] \). Because of special properties of the cepstrum, it is
possible to find a cepstral window which bypasses the computation of \( \log |V(e^{j\omega})| \),
and results in \( \log |V(e^{j\omega})| \) directly.

When this method is used, as usual the time-varying properties of speech require
that an analysis region be used; this means that the input signal \( s(n) \) is multiplied by a
window \( w(n) \). Because the window is reasonably wide, \( s(n)w(n) = (p(n) * v(n))w(n) \approx
(p(n)w(n)) * v(n) \). One would therefore expect that after the processing described
above, at best the result on a single analysis region will be \( p(n)w(n) \), not \( p(n) \). The
fact that the results from each of the analysis regions are windowed in this way would
appear to make it difficult to produce a connected signal, call it \( d(n) \), which unlike
\( p(n)w(n) \) (but like \( e(n) \)) corresponds to the whole input speech signal \( s(n) \). The
obvious suggestion is to overlap the results from each region in just the right way, and
add them together; this means not only that the tails of the windows have to add to
(nearly) a constant, but also that the delay in each of the \( p(n)w(n) \) has to be such
that the impulses in successive regions are synchronized. Neither of these issues was
addressed in Oppenheim and Schafer’s seminal 1968 paper ([77]). However, an answer
to both of them was given as part of the general advances made in the 1970’s in our

---

3 This statement— which was made in [77] and has been repeated many times since—is not very justified; see [124].

4 Another suggestion might be to take the inverse transform of \( 1/V(e^{j\omega}) \), and convolve \( s(n) \) with the
resulting filter. Then \( d(n) \) is calculated just like \( e(n) \), except that the FIR inverse filter is obtained in
a different way. However, this is slower than using the FFT for the convolution.
understanding of Fourier analysis-synthesis systems (e.g. [3], [91]). Under this view, the intermediary stage between analysis and synthesis is represented by \( S_n(e^{j2\pi k/M}) \). For a fixed \( n \), this gives the short-term spectrum of the signal around \( n \), sampled at the frequencies \( 2\pi k/M \). For a fixed \( k \), this gives the outputs of a bandpass filter centered at \( 2\pi k/M \), sampled at the times \( n \). The operation of going from \( s(n) \) to the cepstrally inverse filtered signal \( d(n) \) is done by first finding \( S_n(e^{j2\pi k/M}) \) from \( s(n) \), then modifying \( S_n(e^{j2\pi k/M}) \) by dividing out a smoothed version of itself, and then resynthesizing from the resulting \( D_n(e^{j2\pi k/M}) \) to get \( d(n) \). This smoothing operation could be done in one of two ways: (1) \( S_n(e^{j2\pi k/M}) \) could be sampled finely in \( n \) and sparsely in \( k \); this means that the bandpass filters are wide, so that smoothing in frequency is unnecessary, but smoothing of the power across \( n \) is necessary; this is roughly the method followed by Sondhi; and (2) \( S_n(e^{j2\pi k/M}) \) could be sparsely sampled in \( n \) and finely in \( k \); now no smoothing in \( n \) is necessary, but the spectra (i.e. in \( k \)) need to be smoothed, since the bandpass filters are narrow; this is roughly the method outlined above. (There is a fundamental sampling limit which determines the minimum number of total samples indexed by \( (n,k) \) of \( S_n(e^{j2\pi k/M}) \) that allows reconstruction of \( s(n) \); the number of samples of \( n \) and the number of samples of \( k \) vary inversely with each other.) The two methods of smoothing \( S_n(e^{j2\pi k/M}) \), while not identical, are quite similar, and it is not clear on a theoretical basis why one should be superior to the other. They can both be implemented either with the FFT, or with a filter bank; the resulting four possibilities differ in computational efficiency.

To my knowledge, the specific idea of doing cepstral inverse filtering using short-term spectrum analysis-synthesis techniques has not been reported previously (it is, however, an obvious combination to make, so it probably has been tried). I will not pursue further the subject of exactly how the analysis and synthesis is done; this is discussed quite well in the references cited above.

There are several problems with the cepstrum. First of all, it makes the assumption that the "quefrency" components of \( v(n) \) and \( p(n) \) do not overlap, i.e. that the IDTFT's of \( \log[V(e^{j\omega})] \) and of \( \log[P(e^{j\omega})] \) do not overlap. This assumption is not theoretically justified even for idealized speech signals; in fact, the distance between harmonics for female speech can be greater than the spacing between some formants. In such cases the cepstral inverse filtering will be canceling harmonics as well as formants. The same problem of "throwing out the baby with the bath water" occurs with LPC, the two methods just respond to it differently.

An advantage and disadvantage of using cepstral smoothing for obtaining the inverse filter is that is not based on an idealized all-pole assumption. Because it does not make this assumption, it could, for example, potentially do better than LPC for nasal sounds. However, it is questionable whether this is enough of a reward for being systematically worse at fitting most non-nasal sounds (when LPC is using the right number of poles, preemphasis, etc.).

Because the transfer function is not as accurate, the phase compensation in the spectrum of \( d(n) \) might not be as effective, and so the peaks in \( d(n) \) might be smeared more. Another potential problem with the phase adjustment is that the minimum-
phase compensation is usually done with $V(e^{j\omega})$, which at best includes not only $H(e^{j\omega})$ but also $G(e^{j\omega})$ and $R(e^{j\omega})$. One could try to solve this problem by preemphasizing the spectrum before computing the phase, as with LPC.

There are also problems with using cepstral inverse filtering which are artifacts of the signal processing. When the DTFT is implemented as a DFT, an aliased version of $c(n)$ is computed; this is another reason that windowing out a short time region of $c(n)$ is an operation of somewhat dubious validity. Also, the analysis window $w(n)$ plays a bigger part in $c(n)$ than has been commonly recognized; see [124].

The advantages of the cepstrum, as I see it, are twofold. First of all, it is more flexible than LPC; the spectrum of the inverse filter is directly available, and can be modified easily. Secondly, we observed several problems with LPC which originate from a lack of control over the bandwidths which LPC assigns to its poles. In constrast, the bandwidth of the peaks in the inverse filter is what the main parameter of cepstral liftering—the cutoff of the cepstral window—effectively controls. This means that the bandwidths of the peaks in a cepstral inverse filter are determined more by the cepstral window than by the formants themselves, so that a cepstrum-based technique may not work as well as LPC in ideal cases, but one might expect it to fail more gracefully than LPC in non-ideal cases.

### 2.1.2.2 Simplification

As is apparent from Figure 2.1, the output of the basic transformation typically does not come even close to producing a noise-free sequence of isolated unambiguous pulses with fixed polarity; it is not at all a trivial matter to determine from these signals where the glottal closures occur. These signals are sampled at the same rate as the input speech signal; it would be both foolish and futile to try to use an algorithm which methodically examined every data point. We are thus lead to consider various methods of simplifying the result of the first stage, before trying to find the glottal closure locations, so as to reduce the amount of data that must be considered. The goal at this point is not to in some way improve upon the basic transformation; if that fails on a particular sound, it is probably hopeless to try to salvage something from what it does return. Instead the goal is to preserve as much of the important information as possible, while still reducing the amount of data.

Perhaps the most obvious thing to try is applying a lowpass filter to the signal (be it $e(n)$, or $d(n)$, or something else similar), as Atal and Hanauer did. However, it would be surprising if this were very successful, since $e(n)$ is a zero mean signal; when the smoothing mask is centered around a glottal closure, the large negative spikes and the large positive spikes may cancel each other's contributions, so that the resulting dot product won't necessarily be any larger than in comparatively inactive regions. Also, of course, all low frequency components in the residual will be greatly emphasized. These intuitions are born out in practice; as can be seen in Figure 2.5, linear smoothing is markedly inferior to the other method shown, both in accuracy of the peak locations and in signal to noise ratio.
Figure 2.5: Shown here is an LPC residual, and two different ways of simplifying it: lowpass filtering (convolution with a Gaussian mask .002 seconds long = 6 standard deviations), and squaring followed by smoothing with an anti-causal exponential with mean-life .0006 seconds. The bottom plot shows the glottal closure locations as obtained from an electroglottograph. It can be seen that the lowpass filtering method accentuates large spurious subpeaks in the residual; although it cannot be seen at this scale, lowpass filtering also distorts the locations of the valid peaks. The method of squaring and then smoothing is superior both in accuracy in the peak locations and in signal to noise ratio.
There is a precedent for nonlinear processing of zero mean signals: decoding of AM signals is a well-known example. There too, applying a linear lowpass filter would be of dubious utility, so instead a (nonlinear) peak follower is often used. One might therefore consider, not perhaps a peak follower, but something similar: a squaring operation, followed by smoothing with a narrow mask. Some justification for this idea can be made. If one gives up on the idea of \( e(n) \) being equal to the second derivative of \( p(n) \ast g(n) \), and of \( d(n) \) being equal to \( p(n) \), \( e(n) \) and \( d(n) \) can still be thought of in terms of the predictability of samples of \( s(n) \). One would expect poorer predictability around glottal closures, so that with a running short-term rms computation (which is what the square-smooth sequence does), one might hope to get larger values at glottal closures. As demonstrated in the previous section, this in effect is what Wong et al. ([131]) and Strube ([116]) are doing when they use \( E(n) \), the total rms prediction error for successive short LPC analyses, instead of \( e(n) \). However, doing it directly on \( e(n) \) is simpler, faster, more flexible (an arbitrary smoothing mask can be used), and more accurate (the LPC spectra are not very well estimated with such a short window, unless it is placed just right). Joe Olive and Mitch Marcus at Bell Labs used a lightly smoothed squared LPC residual as part of an inconclusive (actually, abortive) pitch tracking effort (personal communication); however, they used a symmetric smoothing mask. A typical example of the signal resulting from using my smoothing mask is shown in Figure 2.5.

With regard to the smoothing done on the squared signal, it is important to keep distinct three different possible reasons for performing the operation: (1) to remove fricative noise, (2) to facilitate peak finding, and (3) to "unify" the amplitude excursions due to glottal closures, which are spread out by a host of effects. Each of these would tend to suggest different cutoff frequencies. All of them are motivated by a desire to make the simplified signal more robust. The driving motivation for me is the third: it is often hard to tell even visually exactly where the glottal closures are located in the bursts of spikes which occur around them in \( e(n) \) or \( d(n) \). Because of this spreading, one might actually expect a slight smoothing operation to cause an increase in accuracy, over just taking the maximum peak in the vicinity. On the other hand, lowpass filtering generally degrades precision. The question of which of these two effects is dominant will be investigated in Section 2.1.2.4. The simplified signal still has many spurious maxima—about a dozen for every good one—which could have been removed by smoothing the squared signal even more. However, too much smoothing definitely does damage to the resolution, so only enough was done as was thought necessary to unify the excitation bursts, and the remaining spurious peaks were eliminated by nonlinear means—the peak finder, of course, does not shift the peak locations. Some excitation bursts were occasionally found which obviously needed more smoothing than was given, but the smoothing was still kept at the minimum necessary to unify most signals.

The particular convolution mask chosen for smoothing the squared prediction residual is important. What we want is a matched filter for the typical appearance of the squared signal at a glottal closure. Experience with the prediction residual shows that
therefore an even, smooth mask (such as a Gaussian curve) is an not appropriate one; there are almost always more excursions after a glottal closure spike than before. The mask that is used here is an exponential mask with a mean-life of about .0006 sec (decaying after being flipped f r the convolution). This could easily be implemented by a first order IIR system running backwards on the squared residual. That almost always works well, but it was found that sometimes the tail of the exponential caused problems, so instead an FIR truncated exponential .002 seconds long is used. This FIR filter is positioned anti-causally, so that the result of a single dot-product corresponds to the beginning of the mask.

Another method of simplifying the LPC residual that has been proposed by Ananthapadmanabha and Yegnanarayana ([6], [7]), is to take its Hilbert (instantaneous) envelope. They justify this by saying that it counteracts the "effect of the phase angles of formants":

Consider the following representation for a speech signal which includes the amplitudes and phase angles of the formants:

\[ h(nT) = \sum_{i} C_i \exp(-a_i nT) \sin(b_i nT + \phi_i) \]

where \( C_i, \phi_i, a_i/\pi, b_i/2\pi \) represent, respectively, the amplitude, phase angle, bandwidth, and formant frequency of the \( i \)th formant. The amplitude of the LP residual around the excitation instant, i.e., \( nT = 0 \), depends on the phase angles in a complex manner. But the effect of the phase angles can be explicitly seen for the case of a single resonator, with \( C_1 = 1 \), for which the LP residual \( t(nT) \) at \( nT = 0 \) is \( h(0) = \sin(\phi_1) \). Thus, for \( \phi_1 = 0 \), the LP residual at the instant of excitation will be zero instead of an impulse of unit magnitude as in the ideal case. The LP residual \( t_H(nT') \) for the quadrature component of \( h(nT) \) will be \( \cos(\phi_1) \). Thus, for \( \phi_1 = 0 \), the LP residual \( t_H(nT) \) at \( nT = 0 \) for the quadrature component will be an impulse of unit magnitude. However, by computing the Hilbert envelope \( t_0(nT) \), given by

\[ t_0(nT) = [t^2(nT) + t_H^2(nT)]^{1/2}, \]

the ambiguity caused by the phase angle in locating the epoch may be overcome. The signal \( t_H(nT) \) can be computed as the Hilbert transform of \( t(nT) \).

The LP residual \( t(nT) \) and its Hilbert transform \( t_H(nT) \) are shown in Fig. 3 [not reproduced here; see Figure 2.6] for two vowel sounds. Although several other factors could have influenced the LP residual, it can be said that the effect of phase angles \( \phi_i \)s is to introduce bipolar swings near the excitation instant. In the LP residual \( t(nT') \) for vowel /i/ there appears to be a zero crossing at the excitation instant. This contention is supported by the fact that there is an unambiguous peak in \( t_H(nT) \) at the same
location. For arbitrary phase angles \( \phi \), bipolar swings occur both in 
\( t(nT) \) and \( t_H(nT) \), as can be seen for the case of vowel /e/. [7]

This passage is confusing. Every pole has two parameters: its frequency and its band-
width. For speech (or for anything which has a system function with real coefficients),
poles always occur in pairs, which together produce one of the damped sinusoid terms
in the sum given above. Thus there are only two degrees of freedom for each pole-pair,
not four, as the passage would tend to imply; in particular, the phase angles are not
independently specifiable. To change the phases would require adding terms to the
numerator of the system function. There can be a complex amplitude \( A \) for the whole
sum which is independently specifiable, but then \( h(nT) \) is not the impulse response,
but the response to \( A\delta(nT) \). A complex amplitude term would shift the phase of all
the sinusoids by a constant amount, but since the acoustic excitation is real, this is of
no concern. For a single unit amplitude pole-pair (formant, resonator) at \( s = \pm jb - \alpha \),
the impulse response is \( \frac{1}{j}e^{-\alpha t} \sin(bt) \); the phase of this is fixed, not variable. However,
as they want for their example, the phase angle \( \phi \) here is zero; but that doesn’t matter;
the residual \( t(nT) \) is still 1 at \( nT = 0 \), and not 0, as they claim. As proven in [91] and
elsewhere, LPC analysis (of the correct order) will determine the poles of the impulse
response of an all-pole system exactly, so that the inverse filtered signal will necessarily
be an impulse; it is that simple. The inverse filter correctly removes the phase due to
the poles (minimum-phase, the Hilbert transform pair of the log magnitude transfer
function). The bipolar swings in \( e(n) \) can easily be explained, as they point out, by
other factors: noise, poorly modeled high frequency poles, and the double differentia-
tion of \( p(n) * g(n) \). The fact that \( t(nT) \) and \( t_H(nT) \) complement each other is not
support of the formant model given, but is simply a verification of what the Hilbert
transform does: it shifts sinusoids ninety degrees.

Putting aside any justification for using the Hilbert envelope on theoretical grounds,
one might still consider it for its rectifying ability. Consider a very idealized model
of the LPC residual signal around a glottal closure: \( \sin(\omega t) \), where \( \omega \) is the high
frequency at which the signal oscillates back and forth. The Hilbert transform of
\( \sin(\omega t) \) is \( \cos(\omega t) \), so that its Hilbert envelope is \( (\sin^2(\omega t) + \cos^2(\omega t))^{1/2} = 1 \). If
instead, as we have been suggesting, the signal \( \sin(\omega t) \) is squared and smoothed, we
get roughly its rms value, which is also 1. For complicated signals such as \( e(n) \) it isn’t
clear mathematically what either of them does qualitatively; a comparison of them is
shown in Figure 2.6. They are comparable in effect; since squaring and smoothing is
faster, more flexible, and easier to understand, the Hilbert envelope is not used here.

2.1.2.3 Peak Finding

The purpose of the third stage is to find the peaks in the simplified signal which
correspond to glottal closures. Its operation is split into two parts: first find all the
maxima—the output of the simplification stage is assumed to be positive—and then
eliminate the maxima which do not belong. If one were to process the signal with an
absolutely causal ("deterministic," "left to right") algorithm, then these steps could of
Figure 2.6: Shown here is a bandpass filtered LPC residual, its Hilbert transform, its Hilbert envelope squared (residual squared plus transform squared), and its square. The complementary behavior of the residual and its transform is evident. It is clear that the Hilbert envelope squared is very much like a slightly smoothed version of the square.
course be done simultaneously as one is moving through the signal. Such an algorithm is not used here, and since we don’t want to ever have to search for the maximum of a single peak more than once, all of the maxima are found before anything else is done. This first part of determining the peaks in the simplified signal is really another stage of simplification.

The first step of eliminating bad maxima is simple; whenever two maxima are less than 1 msec apart, the one with smaller amplitude is removed. This cuts down on the number of maxima caused by noise, and it partly solves the problem that some peaks may have more than one relative maximum in them. Since 1 msec is much less than a possible pitch period length, there is little danger of removing everything from a veridical glottal closure location.

The new sequence of maxima is used as the input to a somewhat complicated procedure which labels each maximum as either valid or spurious. The sequence of maxima is represented as an array of pairs \((t, a)\), where \(t\) is the time of the maximum and \(a\) is its amplitude. The algorithm is similar in spirit to many others, such as that used by Reddy ([92]); it is much like Specker’s “postprocessing” technique ([110]) and to an algorithm Barbara Caspers of Bell Labs used in some unpublished work on pitch detection from multi-pulse LPC excitations. A Boolean array is maintained which corresponds to the maxima array and indicates at each element whether the maxima with the same index is valid or not. This array is initially all “false”; i.e., all maxima are assumed to be spurious. First, within a short interval (following the previous iteration’s region) an “island” peak (maximum) is found, which appears to be a valid one, and an initial guess of the period around that island peak is made. Then an attempt is made to “grow” out from this peak, by looking for a neighboring peak whose distance is roughly the estimated period. If that attempt is successful, then another attempt is made at finding the next peak (based on an updated period estimate), and so on, in both temporal directions. When it is impossible to find a peak in a predicted interval, or when a peak with a larger amplitude than the ones being kept is skipped, or when successive peak amplitudes differ greatly in amplitude, the loop is terminated. If the “growth” does not go very far, then all the maxima which had been labeled as “true” by the growth attempt are set back to “false.” Then another island peak is looked for in a short interval following the previous region, if the growth was successful, or just a little past the previous island peak, if it was not, and the whole process is repeated. No voiced/unvoiced decision is necessary—but the finished pitch detector could itself be used for one. No significant assumptions about pitch range are made. Although the algorithm does some backtracking, it is mostly deterministic (left-to-right); no global searches are used. It does not depend on any global computations to set thresholds, as might, for example, have been done to detect silence.

The algorithm can probably best be expressed in pseudo-procedural language:

```
;;;global variables accessible to all routines are an array holding (time,amp) pairs for peaks,
;;;sorted by time, and a boolean array giving the status of each of those peaks
```
\textbf{decide-peaks():}

\begin{itemize}
\item\texttt{;;; start-time is the end of the last successive growth region}
\item\texttt{;;; begin-time and end-time are the interval in which an island peak is found,}
\item\texttt{;;; and from which the initial period estimate is found}
\item\texttt{start-time = 0, begin-time = 0, end-time = 0}
\item\texttt{loop until begin-time > time-of-last-peak:}
\item\texttt{begin-time = max(start-time, end-time - INTERVAL-OVERLAP)}
\item\texttt{end-time = begin-time + INTERVAL-LENGTH}
\item\texttt{island-time = find-island(begin-time, end-time)}
\item\texttt{if find-island failed, go back to loop beginning}
\item\texttt{initial-period = initial-guess(begin-time, end-time, island-time)}
\item\texttt{if initial-guess failed, go back to loop beginning}
\item\texttt{label the peak at island-time as valid}
\item\texttt{count = 1 ;; number of valid peaks in the region being grown}
\item\texttt{repeat twice, first direction = backwards, then direction = forwards:}
\item\texttt{current-time = island-time, current-amp = island-amp}
\item\texttt{loop:}
\item\texttt{if current-time < start-time, exit loop}
\item\texttt{if current-time > time-of-last-peak, exit loop}
\item\texttt{(predicted-time, least-time, greatest-time, skipped-amp-frac) =}
\item\texttt{predict-next(current-time, count, direction)}
\item\texttt{find the skipped peaks between the current peak and the range beginning}
\item\texttt{with least-time and ending with greatest-time}
\item\texttt{find the peaks in the range}
\item\texttt{range-amp = biggest peak in the range, range-time = location of range-amp}
\item\texttt{if no peaks in range, exit loop}
\item\texttt{if range-amp < PRED-AMP-FRAC \times current-amp, exit loop}
\item\texttt{skipped-amp = biggest peak in range or skipped area besides range-amp}
\item\texttt{if skipped-amp > skipped-amp-frac \times min(range-amp, current amp), exit loop}
\item\texttt{if there is a peak closer to predicted-time than max-time is,}
\item\texttt{with amplitude > CLOSER-AMP-FRAC \times max-range, exit loop}
\item\texttt{label the peak at range-time as valid}
\item\texttt{current-amp = range-amp, current-time = range-time, count = count+1}
\item\texttt{if count \leq 3, remove valid status from peaks so labeled, and go back to loop beginning}
\item\texttt{start-time = time of last new peak added}
\end{itemize}

\textbf{find-island(begin-time, end-time):}
\texttt{fail if there are no peaks (maxima) between begin-time and end-time}
\texttt{otherwise return time of maximum amplitude peak in the range}

\begin{itemize}
\item\texttt{This is based on a histogram of the periods in the range, with frequencies}
\item\texttt{of occurrence weighted by the period values.}
\item\texttt{initial-guess(begin-time, end-time, island-time):}
\end{itemize}
isolate all peaks between begin-time and end-time,
   with amplitudes > INITIAL-THRESH × island-amp
fail if only one peak found (from island-time, presumably), otherwise proceed
create a list of the time distances between successive peaks
sort this list
separate this list into groups such that successive elements of a single
   group differ by less than HIST-THRESH times their values
rank the groups by the product of their lengths and their mean value
return mean value of the group at the top of this ranking

predict-next(current-time, count, direction):
   if count = 1
      period = initial-guess
      lower-period = PREVIOUS-LOWER × period
      upper-period = PREVIOUS-UPPER × period
      skipped-amp-frac = PREVIOUS-SKIPPED-FRAC
   if count ≤ 4
      period = distance between the two peaks right before current-time
         which were labeled as valid
      lower-period = PREVIOUS-LOWER × period
      upper-period = PREVIOUS-UPPER × period
      skipped-amp-frac = PREVIOUS-SKIPPED-FRAC
   if count > 4 and current width of growing region < .04 seconds
      period = prediction of straight line fit on previous 4 periods
      lower-period = LINEAR-LOWER × period
      upper-period = LINEAR-UPPER × period
      skipped-amp-frac = LINEAR-SKIPPED-FRAC
   if current width of growing region > .04 seconds
      period = prediction of parabolic fit to periods in an interval
         of width min(growth region width, .1 seconds)
      lower-period = PARAB-LOWER × period
      upper-period = PARAB-UPPER × period
      skipped-amp-frac = PARAB-SKIPPED-FRAC
   if direction = forwards
      return (current-time + period, current-time + lower-period,
         current-time + upper-period, skipped-amp-frac)
   else
      return (current-time - period, current-time - upper-period,
         current-time - lower-period, skipped-amp-frac)

The current values of the constants are:
   INTERVAL-LENGTH .05
| INTERVAL-OVERLAP | .5  
| PRED-AMP-FRAC    | .1  
| CLOSER-AMP-FRAC  | .3  
| INITIAL-TİRESH   | .4  
| HİST-THRESH      | .1  
| PREVIOUS-SKIPPED-FRAC | .6  
| PREVIOUS-LOWER   | .8  
| PREVIOUS-UPPER   | 1.25  
| LINEAR-SKIPPED-FRAC | .8  
| LINEAR-LOWER     | .9  
| LINEAR-UPPER     | 1.1  
| PARAB-SKIPPED-FRAC | 1.2  
| PARAB-LOWER      | .95  
| PARAB-UPPER      | 1.05 |

There are a few additional complications in the algorithm not indicated above. First of all, to determine the range of times in which to look for the next peak, the algorithm above multiplies the predicted period with two constants, one a little less than 1 and the other a little greater than 1. This approach is not very true to reality; a better constraint would be based on the greatest amount of change in the reciprocal of the periods, per unit time. In other words, when the periods are short, it is unlikely that successive periods will differ by more than a few percent, but longer periods can differ by much more. Thus an addition is made to predict-next(), which adds the constraint $|\frac{1}{p_2} - \frac{1}{p_1}| < 10000 \max(p_1, p_2)$, where $p_1$ is the previous period and $p_2$ is the potential period ending in the following range. A second modification has to do with the tests for exiting the inner loop of decide-peaks(). As can be seen in predict-next(), a trade-off is made between a period constraint and an amplitude constraint. When few periods have been found, the next predicted period is just the last one found, there is a wide range of possible periods, and there is a low threshold for failing because of skipping a peak. When more periods have been found, the next period is predicted by a straight line fit, the possible period range is narrower, and the skipped amplitude fraction is higher. Finally, when quite a few periods have been found, the next period is predicted by a parabolic fit to the preceding periods, the possible period range is quite narrow, and the skipped amplitude fraction is quite high—greater than 1. In the algorithm given above, these three modes are exclusive. In actual fact, as many of the modes are tried as possible (just 1 at first, then 2, then all three). If any one of them succeeds, the region continues growing. In this way, for example, if the parabolic prediction is unrealistic, the prediction based on the previous value can salvage the situation. On the other hand, if the previous value prediction fails because of a skipped amplitude, the parabolic mode can save things if the peak in the range is well predicted. All three of the possible cases in which only one of the predictors succeeds, were actually found to occur. In the procedure decide-peaks() as it is given above, first regions are grown backwards, and then they are grown forwards. This
means that the forward growth can benefit from the results of the backward growth, but not vice versa. Thus the third modification is that after forward growing is done, backward growing is attempted again, in case the higher order predictors made possible by the bigger region enable the growth to go farther.

The algorithm is admittedly complicated, but it is fairly robust. The fact that it is complex does not mean that it is slow; this stage is faster than all the others, partly because it deals with so few data points. Typically there are fewer than a 1000 maxima to consider per second (after the ones less than 1 msec apart have been removed).

Note that a short-term analysis pitch tracker (such as one based on correlation) could have been used to provide the initial period estimate for growing from the island, and it could also have been used instead of predicting the next period from the previous few. The reason for not doing this is mostly one of principle, although it is quite justified solely on the basis of simplicity or computation time. As discussed in Section 1.2.1, pitch trackers such as ones based on correlation use a different notion of pitch: what I called "mathematical pitch." This means short-term analysis pitch trackers at best are limited by the conflict between mathematical pitch and production pitch. In situations like the ones listed in that section, the two notions of pitch differ noticeably, and short-term analysis pitch trackers will then be more of a hinder than a help to this algorithm. Thus although it might be a reasonable (though expensive) alternative to the method here, I didn't want to design an algorithm which was dependent upon a short-term analysis pitch tracker, and therefore necessarily as limited as it was. While I don't claim that currently this pitch tracker always handles those problem areas where mathematical and production pitch differ, at least the potential for the ability is not ruled out in principle.

2.1.2.4 Adjusting the Peak Locations

The operations of the previous section at best provide the times of the maxima of the peaks in the simplified signal which correspond to glottal closures. One might consider going back now to the unsimplified signal (the output of the basic transformation), and fine-tuning these times. This could be done by replacing each by the time of the greatest absolute value in the unsimplified signal which occurs within some epsilon distance, say .0005 sec. It was suggested in Section 2.1.2.2 that the simplification process might actually cause an improvement in resolution. Upon testing, there was no obviously preferable choice. Sometimes, the residual has a large spike in a place clearly different from where most of the excitation is, and in those cases the square-smooth operation improves accuracy. Other times, the smoothing shifts the peaks slightly away from their correct locations, while the locations are marked by a large spike in the unsimplified residual. In Figure 2.7, an example f0 contour showing the result of both cases is shown. Although neither choice was uniformly better, it was found that on the average, the algorithm without adjustment is slightly superior (in a small test, the method with adjustment had a 20% greater average absolute error). This stage of processing was therefore eliminated, since it didn't have any net positive
Figure 2.7: Here are shown the \( f_0 \) contour corresponding to the unadjusted peak locations, the \( f_0 \) contour after adjusting the locations, and the \( f_0 \) contour derived from the corresponding EGG signal. The \( f_0 \) contours were computed by the \( 1/t_0 \) method. In the .1 to .2 interval, the unadjusted contour is more similar to the one from the EGG, whereas around .6, the adjusted one is better.
2.1.2.5 Extrapolation

So far, the operations described have been consistent with our stated goal of not trying to do better than the basic transformation. All we have done is eliminate proposed glottal closure locations, we have not moved them or inserted new ones. Also, we have used rather stringent criteria for accepting something as a true glottal-closure-originated peak; we have not tried to ferret out peaks in the middle of noise. However, the transformation will fail in some cases, and the question remains as to what should be done in such cases. LPC residual, for example, is occasionally close to useless on voiced fricatives and on voiced sonorants which have mostly low-frequency energy (nasals, liquids, glides, and back vowels). It is not surprising that it does so; we are trying to find the “beginnings” of the periods, a concept which does not make sense for an arbitrary periodic signal. We are attempting to do this by a method which takes advantage of a special fact about speech: at period beginnings it is usually harder to predict samples. However, some valid speech sounds, such as nasals, have little in their signals which makes the glottal closure moments distinguished; this is why the algorithm can fail even when it is perfectly clear what the period is. Because it is desirable to have an algorithm which is generally useful, it is important that it figure out an answer even when the transformation fails. The method of doing this is by trying to grow each of the regions found in the peak picking stage, by checking short-term correlations in the input speech signal at predicted periods. I call this stage “extrapolation.” Consider the case of extrapolating forwards; going backwards is analogous. First, the last period in the region is found. Then the data points of the speech signal within this period are correlated with subsequences of the same length found at a range of positive shifts. The shift with the greatest normalized correlation coefficient is added to the last peak time in the region to obtain the next glottal closure time. This process is then repeated. One would hope that because each additional peak should be close to a glottal closure, that it would stay that way, but clearly this method is marginally unstable, and the peak times will gradually drift away from reasonable glottal closure locations. For this reason, it important that the preceding stages do as much as possible, to keep the extrapolation regions as short. Another reason for doing this, beyond the overriding issue of accuracy, is speed; the dot-products take a substantial amount of time. Thus by having a multi-stage algorithm, the easy and unambiguous areas can be quickly and accurately taken care of; the other areas will be processed more slowly and less accurately, but time is not wasted unnecessarily plodding through the simple parts, and the unambiguous regions can provide context for dealing with the ambiguous ones.

There are three parameters related to the extrapolation process. Two specify the range of shifts to use, expressed as multipliers for the previous period; they are currently equal to .95 and 1.05. The third is a threshold for the normalized correlation coefficient which determines when the extrapolation will stop; it is currently set to .4.
2.1.3 Tests

The following sections describe data, programs, tests, results, and conclusions. An effort was made throughout (perhaps not an entirely successful one) to follow the recommendations laid out in Section 1.4.4.

2.1.3.1 Data

The following ten sentences were used as the text for the speech data obtained:

1. Variability is a bugbear of those who use version eight.
2. Everything's better with Blue Bonnet on it.
3. Analogies are very similar to metaphors.
4. I've ruined lightbulbs, but only occasionally.
5. Is this new ink of yours indelible?
6. Don't meddle in my business!
7. That zoo is judged by many to be better than ours.
8. A Hamming window is adjusted over the relevant region.
9. Logically, this is the most undesirable day.
10. Where are you going, Eeyore?

These were specially selected to be likely to cause trouble; there is a liberal sprinkling of voiced fricatives, voiced stops, liquids, and glides. The word "Hamming" was chosen to provide an example of strong aspiration. Three of the sentences have intra-sentential pauses, and several of the phrases end in vowels, to encourage the occurrence of vocal fry. There are also a few contexts which were designed to stimulate flaps and glottal stops. Eight adult speakers, four male and four female, were enlisted to read these sentences. Not all of the speakers were native speakers of English, but since the phenomena under consideration are presumably universal, this was not considered to be a vital issue. It is notoriously difficult to get subjects to vary their $f_0$ to any significant degree when reading a list of sentences; this is why two questions and one exclamation were included among them. Another thing that was done to offset this problem was to give instructions to the subjects that they were to select a word (of their own choosing) somewhere near the beginning of the sentence, and put their main emphasis on that word, as if trying to clarify a point. It was intended that by making them put the focus in an unnatural place, it would be possible to evoke a wider $f_0$ variation; this appeared to be effective. Note that it is not crucial that all the speakers said the sentences in the same way, or even that they said exactly same thing.
While talking, the eight speakers were attached to a Synchrovoice "Research Electroglottograph," and the speech signals and electroglottograph (EGG) output were recorded in stereo on an FM tape recorder. This was done in a quiet room, and efforts were made to cut down on the effects of room reverberation. The microphone was suspended at a fixed distance from the mouth. The two tracks of the FM tape were then digitized with a Digital Sound Corporation I/O device, at 16000 Hz, with a 6400 Hz anti-aliasing filter and 16 bit quantization. The speech and EGG signal were digitized individually—by playing one, rewinding the tape, and playing the other—because stereo digitization was not conveniently available. (If it had been available, then the FM tape recording would have been bypassed altogether, and the signals would have been digitized directly.) This means that after digitization the two had to be resynchronized; this was found to be time-consuming but not very difficult. There is a big potential for inaccuracy in getting the delay right, but the delay is not crucial; in testing the pitch tracker, we are not concerned with a systematic constant shift throughout an utterance, but with the variance in the estimated glottal closure times. If the delay between the speech and the EGG were not constant (which it is), then there would be problems, but of course that situation couldn't be handled if stereo digitization were used either.

The electroglottograph measures the impedance across the neck by calculating the rms current of a high frequency carrier wave. The more that the glottal folds contact each other, the smaller this impedance. Various schemes for determining the glottal closure times from the EGG have been proposed: find the maxima in the first derivative, find the first crossing of a fixed threshold, and find the point at which it hits half of its extreme value. The first of these methods is the one chosen here; it is the most commonly used. In fact, the digitized signal was not the EGG signal per se, but its derivative (from an analog first-order differentiator). This was done to avoid wasting quantization space on the large swings in the EGG signal which are caused by voicebox motion.

The EGG signal is fairly unambiguous. Numerous studies, comparing it to the outputs of other instruments (photoelectric and ultrasonic devices; acoustic and mechanical transducers), have demonstrated that when it works at all, it works accurately for determining the glottal closure instants ([46], pp.103-132). An example of a speech signal, the differentiated EGG signal, and the resulting glottal closure markers is shown in Figure 2.8. An automatic procedure was used to obtain the moments of glottal closure from the differentiated EGG signal. All maxima which exceeded a fixed threshold (which happened to be 3000) were taken, and if any maxima were closer than .001 seconds apart, the smaller in magnitude was eliminated. The result of the automatic procedure was corrected, when necessary, by hand. It was apparent by looking at the speech and EGG signals together that occasionally the EGG gave no significant output when it should. This problem is inherent to the EGG; in some "glottal closures," the glottis doesn't really close; this can happen in breathy and high-pitch voices. When the glottal folds don't come together, the impedance measured doesn't drop significantly, and so the event is missed altogether. The hand correction was only to counter occa-
Figure 2.8: Shown are a segment of a speech signal, the synchronized differentiated EGG output, and the resulting glottal closure markers. As is customary, the EGG signal is shown with the polarity opposite from how it was first obtained. Since the derivative is what was recorded originally, the EGG signal itself was unavailable. The dips in the differentiated EGG are due to the glottal openings.

Sional problems with the fixed threshold, when an obviously correct maxima fell just below it, or obvious noise got above it. No effort was made to insert glottal closures where they looked suitable, since the goal was to have a standard more accurate than the human visual system. However, the number of cases where the EGG apparently failed is insignificant compared to the performance of the pitch trackers being tested against it, so this is not cause for great concern.

The final result of all this was approximately 250 seconds of speech data, with a total range of f0 of 50-450 Hz, and an average sentence f0 variation of around a 100 Hz.

Note that the fact that obtaining this data was somewhat inconvenient is a testimony to the potential utility of a pitch tracker able to produce glottal closure times automatically from the speech wave. There are many cases in the laboratory where having an EGG signal (or any other glottographic signal) is impossible (for any speech data from TV, telephone, or radio, and any speech data from field work) or prohibitively inconvenient or expensive (in large speech data bases, for example). There are also many applications (see Section 1.4.2) where it is impossible to assume that an EGG signal is available. While surely a pitch tracker using only the speech signal could never compete with a glottograph for reliability or accuracy, there is a large area where having accurate pitch information is useful, but having glottographic information is difficult or impossible.
2.1.3.2 Software

The programs described here were implemented on a Symbolics Lisp Machine, supported by a Floating Point Systems array processor. Interactive graphic signal examination and manipulation were done with the SPIRE (Speech and Phonetics Interactive Research Environment) software system developed for the Lisp machine at MIT, and with programs written by me. The FPS was used for the LPC analyses, FFT’s, and convolutions. The final pitch detector takes roughly 2 minutes per second of speech, for the speech data digitized at 16000 Hz. The largest amount of time is taken by the first step, the computation of the LPC residual.

In the next few paragraphs, the whole operation of the pitch tracker is summarized, reviewing and detailing the previous sections.

The “basic transformation” used is bandpass filtered LPC residual. The preemphasized speech signal is windowed with a .025 second Hamming window; 14th order correlation LPC is solved by the Wiener-Levinson method. The resulting filter coefficients are convolved with the (still preemphasized) speech signal. This is done 100 times a second; i.e., each .025 second window provides .01 seconds of residual. The prediction residual is then convolved with a zero-phase bandpass filter with cutoff frequencies at 700 and 4000 Hz, and 6dB down frequencies at 400 and 4300 Hz.

Another transformation, cepstral inverse filtering, was also implemented for comparison. This was done by the filter bank summation method, using FFT’s (see [91], pp.303-310). 512 point FFT’s were taken of data windowed by a 512 point Hamming window (.032 seconds), every 128 samples. Each of the inputs to the FFT’s were properly rotated to facilitate resynthesis. This gives the two-dimensional signal representation $S_n(e^{j2\pi k/M})$ mentioned in Section 2.1.2.1. Then the (not complex) cepstrum is computed, and multiplied by a minimum phase cepstral window with a pass-time of .002 seconds (window = constant) and a transition time of .002 seconds (window = quarter cosine). Taking the FFT of this windowed cepstrum gives a smoothed version of the original spectrum, with minimum phase. This smooth spectrum is then divided out of the original spectrum, giving the modified signal represented by $D_n(e^{j2\pi k/M})$. The inverse filtered signal was synthesized from $D_n(e^{j2\pi k/M})$ by Portnoff’s technique ([91]), using another 512 point Hamming window for the interpolation. Unlike the case of LPC, there is no very natural way to determine the appropriate gain of the result of the computation on a single frame. Here, the gain was set so that the inverse filtered spectrum had the same amount of energy as the original one did in its 0-2000 Hz subband. An example of the output is shown in Figure 2.9.

Most of the time, although the LPC prediction residual and the cepstral inverse filtered signal always look different, they are roughly the same in quality. However, there are some systematic qualitative differences between them, consistent with the statements made in Section 2.1.2.1. The cepstral signal is often noisier than the residual, and its excursions near glottal closures are sometimes more spread out. This makes sense, since for most signals LPC is based on a better model of the speech production process. On the other hand, for most of the speech sounds which make
Figure 2.9: Above are shown a section of a speech signal, its LPC residual, the result of cepstral inverse filtering, and the moments of glottal closure as determined from an EGG.
LPC fail, the cepstral inverse filtered speech generally has enough action near closure times that a person can at least visually make out what is going on. However, the cepstral signal is never very accurate for those sounds—usually less accurate than the extrapolation procedure described earlier. Thus for both kinds of sounds, the LPC residual is preferable; for sounds where the model is successful, it gives good salient peaks (more so than cepstral methods), and for sounds where the model is unsuccessful, it conveniently breaks down altogether, instead of damaging the quality of the output by giving very noisy peaks. The LPC residual is also faster to compute.

To simplify the bandpass filtered LPC residual, it is squared and then smoothed by convolution with a mask. The mask used is .002 seconds long and has the formula $e^{-nT/\cdot0006} - e^{-0.002/\cdot0006}$. This is convolved with the squared signal (without flipping the mask) so that the beginning of the mask is synchronized with the output of the dot-product.

The maxima of the simplified signal are extracted; when two maxima have times which differ by less than .001, the smaller one is dropped. This sequence of maxima is sent as the input to the peak detection routine, which eliminates the spurious maxima using the algorithm given in Section 2.1.2.3. The output of the peak detection is an array of the valid maxima times, and a list specifying the endpoints of the regions. These two are given to the extrapolation routine, along with the original speech signal. The extrapolator tries to extend the regions, by testing the short-term correlations in the speech signal at the boundaries of the regions. The final output is then an augmented list of glottal closure times. If desired, this list can be converted to an f0 contour by one of the methods described in Section 1.2.3.

The initial development of the programs was done on miscellaneous speech signals not included in the test set. A total of four runs were done on the test set. The first run made it evident that the more complicated period prediction scheme described above would be necessary, and a single sentence with large f0 swings was selected out of the eighty and "tweaked" until performance on it was nearly perfect. The second run revealed a variety of software bugs, mostly having to do with the extrapolation procedure. The results of the third and fourth runs were nearly identical, and are the ones reported here.

2.1.3.3 Results

The dominant theme of this chapter so far has been accuracy, and it is therefore also the main criterion by which this pitch tracker is tested. A second subsidiary goal is that it be reasonably dependable for laboratory quality speech signals. This goal is subsidiary in that very little sacrifice in accuracy is tolerated in return for improved reliability. It would be easy to take one of the standard pitch detectors from the literature, and fuse it with this one so that together they achieve the same reliability as the one added on, with a concomitant degradation in quality whenever the standard pitch detector makes its contribution. In particular, it would have been quite easy to use the extrapolation procedure by itself, with no initialization from the peak finder,
to produce a pitch contour with fewer omissions (glottal closure markers would be impossible, since the correlations wouldn’t be starting in the right place). But if one were to do something like that, it isn’t clear why one should bother with this tracker at all, except perhaps because it is faster than some other methods. For the same reason, this pitch tracker has no “postprocessing” which goes through the output of the pitch tracker and corrects values by improvising ones more consistent with the context. The primary goal in developing this pitch tracker was to determine whether it is possible to derive glottal closure locations automatically and accurately from a speech signal, and so it restricts itself to only labeling glottal closures when it can be confident in its accuracy. Evaluation criteria other than accuracy and reliability, such as robustness in the face of noise and efficient computation time, are incidental to the effort of meeting these two. Rather than make compromises for the sole reason of shortening the computation time, it was thought better to first ascertain how well one can do without any unnecessary sacrifices. In the end, this tracker is not much slower than most other methods, and it is faster than some.

There are three kinds of inaccuracies associated with testing, which should be distinguished. The first, and most fundamental, is the uncertainty and lack of resolution “in the world”; as pointed out in Section 1.1.1, glottal closures are not instantaneous events, and the folds can contact each other at discontinuous points. A second kind of inaccuracy is in the standard for the testing, in this case the electroglottograph. There are many contributing factors here: capacitance between the contacts and the conductors underneath them, electrical interference, acoustic/mechanical effects, inaccuracies in the principle of its operation, etc. All means of measurement are subject to these or similar problems; the basis for calling one measurement device more or less accurate is physical intuition (its hard to beat stroboscopic film) and consensus between different measurement schemes. Some of the more exact devices (by these criteria) are invasive and limit the freedom to speak—so comparison is limited to a rather small subset of the circumstances in which it would be desirable. Despite these observations, it is still possible to state with a fair amount of certainty that the cumulative noise caused by the first two kinds of error never comes to more than a jitter of about .1 msec ([46], p.507). The last kind of inaccuracy is the one that concerns us here: inaccuracies in the pitch determination algorithm. Pitch trackers are effectively limited in their accuracy by the first two kinds of inaccuracy, and fundamentally limited by the first.

The major test of the tracker is a comparison of the estimated times of glottal closure that it provides from speech signals, and the times of glottal closure obtained from the EGG signals corresponding to the same speech. This was done for all of the data described above. A histogram of the signed differences between the times of each of the EGG glottal closures, and the times of the nearest estimated glottal closures, is shown in Figure 2.10. The results are shown for the algorithm running both with extrapolation, and without. The difference between the two histograms would be the histogram of the errors due to only glottal closures found from extrapolation. If the extrapolation procedure quickly started drifting away from the true period beginnings, one would expect to see a uniform distribution a half a pitch period wide for the errors
Figure 2.10: Shown is a histogram of all the signed differences between the EGG closure times and the nearest estimated closure times, measured in seconds. Large differences, due to deletion errors, are not shown. The errors for the unextrapolated closures are plotted in the thin line, and the errors for the unextrapolated and extrapolated closures are plotted in the thicker line.

due to extrapolation. As can be seen here, the drift only gets to be about ± .2 msec; this is probably because the extrapolation only extends the regions short distances.

To compute the pooled histograms shown, it was necessary to first remove the offsets in the errors for each sentence caused by human error in synchronizing the EGG with the speech. This was done by subtracting the means of the errors for each sentence, from the errors obtained from that sentence. Typically these offsets were on the order of a a millisecond, much larger than the resolution of the algorithm. Because of the removal of the means, it is impossible to discover any systematic bias in the estimated times, but that is a much less important (and smaller) parameter than the spread in the errors.

The errors can be classified into three categories: fine errors, which occur when there is an estimated closure time within .0015 sec of an EGG time; deletion errors, which occur when there is no estimated time within the fixed range of an EGG closure time; and insertion errors, which occur when there is no EGG time within the range of an estimated time. As can be seen from the histogram, a threshold of .0015 seconds is not unreasonable for distinguishing fine errors from deletion errors. Every estimated glottal closure time produces either a fine error or an insertion error, and every EGG

---

5Actually, the mean wasn't used because it was too much influenced by outliers. The most common occurring value (within an epsilon) was used instead.
glottal closure time is associated with either a fine error or a deletion error. The cumulative result on all of the data, without extrapolation, was 17246 fine errors, 175 insertion errors, and 6242 deletion errors. With extrapolation, there were 20876 fine errors, 1975 insertion errors, and 2612 deletion errors. Thus the total success rate (fine errors against all errors) was 72.9% without extrapolation and 82% with it. The total recognition rate (fine errors against fine and deletion errors) was 73.4% without extrapolation and 88.9% with it. The average standard deviation (across sentences) in the fine errors was .125 msec without extrapolation and .243 msec with it. The pooled standard deviation (after removing offsets in the errors as for the pooled histogram) was .15 msec without extrapolation and .274 msec with it. Since the sampling period was .0625 msec (the rate was 16000 Hz), a substantial number of glottal closures were found within two samples, without extrapolation. This resolution is of the same order of magnitude as that of the EGG, at least for the way that the EGG was processed here.

There were very few insertion errors without extrapolation, since the requirements of the peak finder are so stringent (they have to be, to be accurate). The insertion errors with extrapolation were caused by there being too high a correlation with whatever was surrounding the voiced speech. Raising the threshold for the correlation coefficient increases the deletion errors. Some of the apparent insertion errors can be attributed to failure of the EGG. The deletion errors were usually caused by problems in the basic transformation. Voiced fricatives and nasals were the biggest problem for the algorithm without extrapolation; back vowels, liquids, and glides did not cause as many problems. Usually the extrapolation procedure managed to fill in these bad spots, but when they were bounded by sudden segment transitions, it was impossible for the extrapolation to get started, because the initial correlations were too low and too noisy. The peak picking algorithm was found to be quite effective whenever the transformation did not produce too egregious results, but it occasionally failed to track sudden or chaotic changes in the period. The parabolic prediction scheme allowed it to follow some quite swift changes, however. It usually succeeded on vocal fry and glottal stops, as long as the pattern of the periods was not too irregular. As can be seen in the algorithm decide-peaks() in Section 2.1.2.3, it is impossible to detect voiced regions with 3 or fewer glottal closures with this algorithm. It is difficult to see how this could be fixed, since there are so many large peaks caused by noise; perhaps it could be done if some better characterization of noise were included. Also, currently the algorithm categorically fails on paired pulsing (a short long short long etc. sequence of periods), because the period continuity constraint is always violated. This could probably be fixed by checking “double period continuity” as well, and accepting the peaks if that succeeds. The performance on male speakers and on female speakers was about the same—this is important to note, because higher pitched voices have more glottal closures, and therefore contribute more to the error measures. One might have expected the algorithm to perform better on female speech, since the higher f0 means that there would be less variation from period to period, but this effect must have been counteracted by the fact that the prediction residual is better behaved for
male speakers.

It is difficult to know how to interpret these results, for several reasons. First of all, to my knowledge this is the first quantitative comparison between a glottographic signal and a signal derived from the speech waveform (automatically or not), and so therefore there is no precedent to compare it to. Boves and Cranen ([16]) came the closest to doing this, when they compared the result of Wong et al.'s inverse filtering ([131]) with three glottographic signals. However, they did not specify any quantitative data, and one of the major conclusions of the paper was that the procedure that Wong et al. followed could not be automated or generalized. Hess and Indefrey ([47]) discuss and recommend the use of a laryngograph in testing pitch trackers, but they only present results on the very preliminary problem of processing the laryngographic signals themselves.

Instead of vainly searching for grounds for a comparison in the past literature, a ballpark idea of how accurate the algorithm is can be obtained by comparing it to what an ideal pitch tracker based on waveform primitives would do; for every glottal closure time indicated by the EGG, the closest zero crossing (or whatever) in the speech wave is found, and this provides a set of error differences. The result of this for a single sentence is shown in figures 2.11 through 2.14, for positive going zero crossings, negative going zero crossings, minima, and maxima. Since usually pitch trackers which use waveform primitives lowpass filter the speech before processing it, the results are also shown for speech which has been lowpass filtered to around 1000 Hz (by convolution with a 2 msec Hamming window). These results are worse, since there are fewer miniscule primitives available to “cheat” the evaluation. All of these are from a single 3 second sentence, spoken by male speaker KS: “Variability is a bugbear of those who use version eight.” As one would expect, the better histograms reveal a typical error of around .001 seconds—half the period of a 500 Hz formant. Of course, it is very generous to suppose that one of these pitch trackers would actually take the nearest waveform primitive to a closure, since continuity and amplitude constraints would probably lead it to choose others nearby instead. From these histograms it possible to conclude that the accuracy of the algorithm given here is about an order of magnitude better than the accuracy any method based on speech waveform primitives could even hope to obtain.

It is also interesting to compare the algorithm with the performance of a person trying to label glottal closures by hand. I went through the same (unfiltered) speech signal as for the previous comparison, and marked where I thought closures were, using an interactive graphics program and a mouse to point at places on the display. This test is, of course, very dependent on the speed with which the hand labeling is done, and it is also affected by the speech used, by the quality of the interactive aid, and by the knowledge of the person doing it. In this case, the labeling was done rather quickly; less than one second was spent per glottal closure.⁶ The result

⁶When the computers available are slow at signal processing but have good interactive display software, pitch detection can be done both more quickly and more accurately by hand than with many pitch
Figure 2.11: Shown is a histogram of all the errors for an idealized pitch detector based on zero-crossings. The errors for positive going zero crossings are in the thin line, and the ones for negative going crossings are in the thick line.

Figure 2.12: Shown is a histogram of all the errors for an idealized pitch detector based on waveform extrema. The errors for minima are in the thin line, and the ones for maxima are in the thick line.
Figure 2.13: This is just like Figure 2.11, except the speech was lowpass filtered first.

Figure 2.14: This is just like Figure 2.12, except the speech was lowpass filtered first.
Figure 2.15: This is a histogram of the fine errors made in human labeling, when compared to an EGG.

is shown in Figure 2.15; although the hand labeling is worse, it isn’t terrible. If a more sophisticated system were used, the result would probably be comparable to the automatic program in accuracy, and undoubtedly it would be more reliable. Such a system could, for example, display the LPC residual instead of the speech, and instead of just accepting the location pointed to with the mouse, it could find the location of the biggest excursion near there. The purpose of this comparison was mostly to demonstrate that human accuracy is by no means an upper bound on the accuracy attainable automatically (unless, of course, the human just simulates the program).

Beyond the fact that there is little precedent for using a glottographic standard, a second problem in interpreting the results is that these speech signals are particularly difficult to handle. Most tests of pitch trackers in the past (when they have been done on laboratory quality signals) have been more lenient. In addition to the speech being difficult, the EGG is a hard taskmaster; it reveals some regions of speech as being voiced which might well be considered unvoiced by the usual standards for pitch tracker testing (visual inspection, semi-automatic methods, listening tests, etc.).

Now if this last problem were the only one, its solution would be obvious: simply try out other pitch trackers on the same data. However, this brings out a third problem, which is that whether or not they have been tested with a glottographic signal, many pitch trackers are based on a completely different notion of pitch, and the notion of error rate in them is completely different. Some pitch trackers, the time-

determination algorithms. The interest in automatic procedures is, of course, that humans can do more important things with their free time than computers can with theirs.
domain ones, can be compared to this one by idealizing them, as we saw above. But the preponderance of pitch tracking schemes have to do with mathematical pitch, not production pitch; they seek to find only local periodicity. Finding the local period is a much simpler problem than finding glottal closure locations (period beginnings); the better error rates for mathematical pitch trackers reported in the literature are around 5%. Therefore a comparison with these trackers is not terribly informative.

Mostly for curiosity, a comparison was done between an f0 contour generated from estimated glottal closure times, and an f0 contour obtained with the mathematical pitch tracker described in Section 2.2. The same sentence was used as for the preceding small tests, and the f0 contour generated from the EGG labels was used as the standard. All f0 values within 10% of the value of the EGG-derived f0 contour were considered to be fine errors. The standard deviation of the fine errors for the production pitch tracker was 2.09 Hz without extrapolation (2.4 Hz with it), and the standard deviation for the mathematical pitch tracker was 2.95 Hz. The f0 contours were computed by the 1/t0 method, and all contours were sampled at 500 Hz. This result is roughly what one would have expected, at least for the production pitch tracker; the location times are typically off by around .2 msec, and the period of this speech was around 10 msec, so the error in the estimate of f0 was around 2 Hz. To emphasize the statement made in the preceding paragraph, these results are not very important, because in the applications which don't require accuracy (e.g. visual displays) they are irrelevant, and the applications which do require accuracy are usually better done with glottal closure locations, not with f0 (see Section 1.4.2). Since mathematical pitch trackers only give the period, and not the glottal closure locations, the comparison is not of much significance. In addition to being unimportant, the comparison is also close to meaningless, because the errors are on the order of the differences between various reasonable ways to compute f0 from closure times (see Section 1.2.3).

---

7For the reasons outlined in Section 1.4.4, this may not mean that the pitch tracker would work that well on this data.
2.2 A Mathematical Pitch Tracker

As stated in Section 1.4, and for the reasons given there, pitch trackers of mathematical pitch are fundamentally limited in their capability to provide useful information about speech signals, and we have probably by now come close to that limit; whether or not one finds this limit confining depends on the uses one intends to put pitch trackers. Most of the algorithms discussed in Section 1.1.2.1 can be converted into fairly reliable pitch (quasi-periodicity) detectors for speech signals. They do vary somewhat in computation time, reliability, and robustness against noise, but their rough performance characteristics are about the same when they are treated with the same amount of developmental effort.

There is thus little need for another member in the family of short-term analysis pitch determination algorithms. However, there are three reasons that this one has been implemented anyway. First of all, it was desired to have a comparison between the \( f_0 \) contours generated from the event times of production pitch, and \( f_0 \) contours taken as the output of a pitch detector of mathematical pitch; the results of this comparison were given in Section 2.1.3.3. Secondly, although the definitional faults of the algorithms in Section 1.1.2.1 do not necessarily have any pernicious effect on their performance with speech signals, it would be nice to have an algorithm which was free from those faults, and which could be easily used (with no or few modifications) on non-speech input. Thirdly, there are several interesting questions which could be investigated with this pitch detector, such as how harmonics behave in a short-term spectral representation when the fundamental is changing.

The description given here is rather abstract and general for two reasons. The first is quite simple: the algorithm described here, while functional, has not been tweaked or tested much, so that it is difficult to say with much confidence what any of the specifics should be. The second reason is that because there are already so many short-term analysis pitch detectors, most of which are no worse and no better than this one, it would probably be most useful to introduce this pitch detector in a way which illustrates issues which come up generally in the implementation of any short-term analysis pitch detector.

The most obvious implementation of a pitch tracker based on the definition of quasi-periodicity given in equation 1.4 proceeds by going through a sequence of three operations at regular intervals in the speech signal: (1) perform a short term spectrum estimate, (2) find the harmonics (if any) in the computed spectrum, and (3) combine the harmonics to arrive at an estimate of the "short-term" quasi-fundamental of the signal, valid in the neighborhood of the analysis region. There are several pitch trackers which have used this general approach in the past; they are sometimes referred to as "harmonic selection" algorithms. In the next three sections I will give an overview of some of the issues and problems that arise in each of the stages; I will discuss how these have been dealt with previously; and I will describe how the stages are implemented in my own algorithm.
2.2.1 Spectrum Analysis

The goal of the first stage is to compute a short-term (but narrow-band!) frequency representation which facilitates the extraction of harmonics by the second stage. While definition 1.4 as given is completely general, it is necessary to introduce several additional assumptions—properties special to a subset of all quasi-periodic signals—in order to use it. It is simplest if these parameters are fixed in the pitch tracker for a particular large class of inputs (say, high-quality speech signals), but it may become necessary to change these for different inputs (say, male vs. female speech), or even to change them during the processing of a single input (say, with changing pitch).

One of these parameters is the range of frequencies which might contain harmonics, i.e., the range of frequencies at which the spectral representation should be computed. For high-quality speech signals, harmonics can be resolved in roughly the range of 0-5000 Hz; telephone-bandwidth speech contains only the harmonics in the range 300-3000 Hz. Musical instruments would have a variety of different ranges. If the basis of the transform is to be the FFT, then the most natural range to use is from 0 Hz to half the sampling rate of the signal. However, people have used various techniques for obtaining other than this range, for example by using the chirp z-transform ([101]), or by using "fft pruning" ([111]).

The spectral representation must be sampled finely enough to be able to separate the harmonic peaks. To be able to avoid aliasing, say, a spectrum which has harmonics at frequencies \(nf_0\), for all \(n\), it will certainly be necessary to sample it at least at the rate \(2/f_0\) samples/Hz, and it would be much better for the second stage if the rate were around \(4/f_0\). If a fixed spectral sampling rate is to be used, it must then be greater than roughly \(4/f_{\text{min}}\), where \(f_{\text{min}}\) is the smallest expected fundamental.

There are many spectrum estimation techniques; these include those based directly on the Fourier transform (periodogram methods, for example), on linear prediction ("maximum entropy" estimation), and on nonstationary signal representations such as the Wigner transform ([20]). By far the most common method in pitch detection has been to simply use a monotonic function of the short-term Fourier transform of the signal. This is what I use as well; it has the advantages over the others that it is fast to compute, it is perspicuous, and it has a fairly well-developed theory behind it. All of the methods require a specification of an analysis window, which determines the size and location of the signal region over which the computation is to be done. This requirement, which is inherent to the idea of a "short-term" calculation, is the source of several difficulties. Clearly it is necessary to have at least two periods within the analysis window; i.e., it must be at least \(2/f_0\) seconds long. If a fixed width is used, it must be greater than \(2/f_{\text{min}}\). However, the width cannot be too great, because everything within the analysis window is used to arrive at a single spectrum, and therefore any changes that occur within the window will be averaged into the result. This of course will affect the accuracy; ultimately if the analysis window becomes too wide the distortions will make the spectrum unintelligible. Although with a definition of pitch based on underlying phenomena (as described in section 1.2), it might make
sense to talk about periods even when successive period shapes are drastically different, both intuitively and by definition 1.4, quasi-periodicity is impossible when properties change at a rate comparable to the "periodicity." If it were often not possible to satisfy both the lower limit on the width, required by getting a certain number of periods in the window, and the upper limit, required by the changing signal properties, then the signal would not be quasi-periodic. It is because it is usually possible to meet both these requirements that signals such as speech are considered to be quasi-periodic. Things are still not easy, of course. Sreenivas ([113]) delineates four different kinds of aperiodicity for speech signals: (1) noise, either from frication or from non-speech sources, (2) changing period lengths, (3) changing amplitude (period size), and (4) changing transfer function (period shape). All of these contribute to the distortion of the spectrum even for "steady state" regions, no matter how small the analysis width. Also, there will still be occasions where one of these factors changes suddenly and significantly (faster than a period length), thus making the signal non-quasi-periodic: in segment transitions (3 and 4) and in glottal stops or paired pulsing (2).

I was just arguing that most of the time there will be some analysis window appropriate for a region of a quasi-periodic signal. This does not mean, of course, that there is necessarily a fixed window width which will work for all regions of the signal; it is conceivable that the rate of change of periods might scale with the fundamental. It so happens that for speech signals this is by and large not the case; i.e., successive period shapes and lengths are more likely to be similar for smaller periods. It thus does not matter that for high fundamentals there might be five times as many periods within a fixed analysis window as for low fundamentals, because the amount of variation within the window will be roughly the same. A wide range of fixed analysis widths have been used for speech in the past; they are almost always somewhere in the range 20-80 msec.

The criterion of having the spectrum finely sampled determines the number of data points in the output of the FFT, and the criterion of having a suitable analysis width determines the number of data points in the input. Since the number of input and output samples in an FFT are equal, it would appear that we have a problem. This is not the case. In the short-term Fourier transform, a shifted copy of the input signal is multiplied with an analysis window, a low-pass signal with a finite number of non-zero points, before the DFT is taken ([91]); the window might be a truncated Gaussian, for example. Clearly the data under the tails of the window play a much reduced role in the resulting spectrum; the quantity that I have been referring to as "analysis width" is really the "effective width" of the window, not its total width. The total width is what determines the order of the FFT, and it can be increased by either extending the tails of the window, or by zero-padding. It can shrunken by a method known as "time-aliasing" (see [91] for an explanation of all these techniques). Another reason that this is useful is that generally FFT's can only be taken on certain sequence lengths, often only lengths which are powers of two.

Both of the criteria just mentioned entail a specification of an \( f_{\text{min}} \), the least possible quasi-fundamental. As the quasi-fundamental becomes smaller, the analysis width and the frequency resolution must increase inversely, so we cannot allow arbitrarily
low fundamentals with this method. People sometimes can produce pitches which are quite low, 20-30Hz, in vocal fry or around glottal stops. However, this cannot be maintained as a "steady-state" excitation; for this tracker, I have assumed a higher minimum of 60 Hz or so, since that is lower than the lowest rate at which nearly anyone can hold their laryngeal vibration steady.

Beyond the distortions in the spectrum due to the aperiodicities in the signal within the analysis window, the signal processing itself causes some distortions. Even if a periodic signal is used as the input, the short-term spectrum computed will not be the ideal line spectrum. As is well-known, the operation of multiplying a signal with an analysis window before taking the Fourier transform is equivalent to convolving the Fourier transform of the signal (a line spectrum for periodic signals) with the Fourier transform of the analysis window. This has a couple of consequences. Because of the smoothing operation the location of the peaks in the spectrum will be shifted slightly. Note that this is not necessarily the case; if the input signal is a periodic impulse train, then its Fourier transform will also be an impulse train, and so none of the peaks will be shifted if a convolution is done with a symmetric window transform. However, the problem will arise for nearly all other signals. Another problem is that most commonly used analysis windows have transforms which are not monotonic, but have numerous subpeaks, so that the convolution operation will introduce spurious peaks.\(^8\) Although the subpeaks in the transforms of most analysis windows are many dB down from the magnitude of the transform at zero frequency, the variation in the amplitudes of genuine harmonic peaks of a single spectrum might be equally large, which means that special processing will be required in the second stage if it is necessary to find the low amplitude harmonics. It is partly for these reasons that Atal ([46] pp.393-396) does not use Fourier transform methods to compute his spectrum; instead he fits a 41th-order LPC analysis to the speech after it has been downsampled to a rate of 2 kHz. Since so many poles are available, the LPC optimization will assign poles to the individual harmonics. With this method, finding the harmonics after the "spectral" computation is not hard: root the polynomial (he used the Newton-Raphson technique), and select the poles with a narrow bandwidth. He avoids both the problems of spectral quantization and of spurious subpeaks.

### 2.2.2 Finding Harmonics

The next stage of the computation takes the computed spectrum and determines from that the locations of the harmonics. A direct application of definition 1.4 would suggest that the harmonic locations are exactly those locations of the spectrum which are nonzero. The observations of the previous section demonstrate that in practice

\(^8\)Monotonicity is a necessary but not a sufficient condition for avoiding the introduction of spurious peaks by convolution. It turns out that the only convolution kernel which is guaranteed not to bring in extra peaks for any signal is the infinite Gaussian ([11]).
this will not be the case. The occurrence of noise, aperiodicity, and convolution with
the analysis window transform implies that the spectrum is almost never zero, that
the harmonics are not just impulses, and that not all maxima in the spectrum mark a
harmonic location.

As in the previous stage, it is necessary to introduce a few assumptions special
to the particular set of inputs under consideration. One of these is with regard to
the relative magnitudes of the harmonic peaks. For speech, there is considerable
variation; although the harmonic magnitudes of the acoustic flow signal at the glottis
fall off gently at roughly 12dB per octave, the resonances of the vocal tract shape the
spectrum enough that salient harmonic peaks may differ by as much as 40 or 50 dB.
Thus many pitch detection schemes which depend on finding harmonics use the log
magnitude DFT spectrum of the speech rather than the magnitude. Sometimes the
fourth root spectrum is used because of the unfortunate behavior of the log function
when its argument is close to zero (\cite{113}).

A variety of methods have been used to recognize the harmonic peaks in the spec-
trum; all of them involve some kind of nonlinear, structural pattern recognition. Senef
(\cite{101}) starts with a list of all the maxima in the spectrum, and associates with each
of them the area of the peak in the magnitude spectrum (defined by two valleys). She
then eliminates all maxima (peaks) which are within 40Hz of a larger peak, and she
also eliminates ones within 67Hz if the peak area ratio is more than 2. The algorithms
described in \cite{4}, \cite{113}, \cite{119}, \cite{29}, and \cite{79} use similar techniques on the magnitude,
log magnitude, or fourth root spectrum; some of the methods besides Seneff's are
a symmetry test, removing the copies of the transform of the analysis window (and
hence the subpeaks) from the spectrum as the peaks are found, using a finite state
automaton for the pattern recognition, and checking the phase characteristics of the
peaks.

Since one can fail to recognize a substantial fraction of the harmonics of a spec-
trum, and still retain enough of them that they have the same fundamental, this stage
is not overwhelmingly difficult. The method used in the algorithm here is to smooth
the magnitude spectrum with a triangular window of total width 60 Hz, and take the
5 highest maxima in the result. No doubt one of the more intricate methods would
be necessary to make it very reliable, but it works surprisingly well. Quadratic inter-
polation is done with the two samples of the spectrum on either side of the maximum,
to improve the accuracy of the estimate.

2.2.3 Combining Harmonics

Numerous methods have been proposed for combining the harmonics to estimate
the fundamental. Seneff (\cite{101}) uses the most frequently occurring difference between
successive harmonic frequencies, using a threshold for two differences to be considered
"equal." Parsons (\cite{79}) uses a method similar to the harmonic sum spectrum (see Sec-
tion 1.1.2.1), which in turn is similar to Schroeder's period histogram (\cite{98}). Amuedo
([4]) uses a more complicated technique based on the same principle. Duifhuis and Willems ([29], [106]) check all possible fundamentals in a quantized range (50-500 Hz), by seeing how many of the harmonics come within a small distance of a multiple of the tentative fundamentals. They avoid the subharmonic matching problem by eliminating fundamentals which have fewer than half of the first 11 multiples present as harmonics (this would clearly be problematic for nonspeech signals with few harmonics). Terhardt ([119]) checks each of the potential fundamentals \( f_s/n \) for consistency with each of the harmonics, where \( n < 12 \) and \( f_s \) is the frequency of the lowest harmonic frequency. He avoids the subharmonic matching problem by weighting higher fundamentals preferentially.

All of these methods are either not fully general, or they deal with the subharmonic problem in a not very clean way. Of course, the methods may work just fine on speech signals, but it would be better if there were an algorithm which dealt with the subharmonic problem without requiring somewhat ad hoc methods which only work for a subset of the periodic signals. Terhardt's algorithm is motivated by psychoacoustic data, but as discussed in sections 1.1.3 and 1.3.1, such data (although important and interesting) are not very relevant to the project of finding mathematical pitch, which is the subject we are concerned with here. One approach to combining harmonics, which is both general and does not have the subharmonic matching problem, is to use a quasi-gcd algorithm, as suggested in Section 1.1.2. This is the approach taken by Sreenivas ([111], [113]).

The Euclidean algorithm ([53]) is an efficient algorithm for finding the gcd of two integers. It cannot be directly applied to the problem of determining the fundamental from estimates of the harmonic frequencies, for two reasons: first, the harmonic locations are real numbers, not integers, and second, they are never found exactly, because of the various effects discussed in the previous section. The first of these problems is easily taken care of, since the definitions of divisor and greatest common divisor can be extended to the real numbers with no revision:

\[
x \text{ is a divisor of } y \text{ if there exists a } k \text{ in } \mathbb{Z} \text{ such that } y = kx.
\]

The notation for this relationship is \( x \mid y \).

\[
g = \gcd(x, y) \text{ if (1) } g \text{ is a divisor of both } x \text{ and } y, \text{ and (2) any number which satisfies } (1) \text{ divides } g.
\]

(2.6)

For integers, \( g, x, \) and \( y \) are in \( \mathbb{Z} \); for reals they are in \( \mathbb{R} \); only \( k \) must be in \( \mathbb{Z} \). The basic properties of the gcd hold for both integers and reals; for example, it is commutative, associative, and \( \gcd(x, y) = \gcd(y, x - ky) \). Note that an integer \( y \) has a finite number of (integral) divisors, and it takes a little work to find them, while a real \( y \) has an infinite number of divisors, but it is trivial to specify them: \( y/n \), for all (positive and negative) integers \( n \). Two integers always have a common divisor, namely \( 1 \); for integers, if \( \gcd(x, y) = 1 \), we say \( x \) and \( y \) are relatively prime. Two reals do not always have a common divisor, in which case they certainly don't have a gcd; the numbers \( \pi \) and 5 are an example. (We could say that two real numbers are relatively prime if they have no common divisors, and set \( \gcd(x, y) = 0 \); this occurs
whenever their ratio is irrational.) The gcd for two integers is the same under either
definition. For our purposes, condition (2) should be rephrased to say "g does not
divide any number which divides x and y, unless that other number also divides g."
This makes no difference for integers, for which the gcd is unique up to sign, but it
does matter for some of the later sets we will consider.

The second problem, that the harmonics are not measured exactly, is more diffi-
cult to take care of. The Euclidean algorithm can't be used unmodified because,
for example, it would give .1 as the gcd of 100 and 200.1. Actually, it is more than
just a measurement problem; real numbers cannot be represented with perfect ac-
curacy on a computer, and so all real numbers have a trivial common divisor of 1e-n,
where n is the computer precision. Although there has been some work on com-
puting the quasi-gcd of polynomials ([103]), Sreenivas is the only one I know of who has
considered the problem for real numbers; the results for polynomials are not directly
applicable. Sreenivas attempts to solve the measurement problem by introducing an
algebra of pairs (A,a), where A is the estimated harmonic frequency, and a > 0 is
the uncertainty in its measurement (i.e., the actual number is somewhere in the range
A ± a):

\[
(A, a) \pm (B, b) \equiv (A \pm B, a + b) \\
k(A, a) \equiv (kA, ka) \\
(A, a) \sim (B, b) \text{ if } |A - B| < (a + b) \\
(A, a) \text{ divides } (B, b) \text{ if there exists an integer } k \text{ such that } k(A, a) - \\
(B, b) \sim (0, 0).
\]

He uses the same definition of the gcd as in definition 2.6. He then applies the usual
Euclidean algorithm with this new algebra, and terminates it when the remainder is \sim
(0, 0). The Euclidean algorithm depends on the fact that gcd(X, Y) \sim gcd(Y, X - kY),
i.e. that G|X and G|Y iff G|Y and G|(X - kY), and this property holds for his
definitions of +, -, \sim, |, and scalar multiplication. Thus, for example:

\[
gcd((1510,10),(700,10)) = gcd((700,10),(1510,10)-2(700,10)) \\
= gcd((700,10),(110,30)) = gcd((110,30),(700,10)-6(110,30)) \\
= gcd((110,30),(40,190)) = gcd((40,190),(110,10)-2(40,190)) \\
= gcd((40,190),(30,390)) = ...
\]

A problem becomes evident when one attempts to terminate this loop, and arrive
at a particular real number as the gcd. In this case, Sreenivas would have stopped
with the answer (110,30), since (40,190) \sim (0, 0). The multipliers suggested by this
answer are 6 and 14, since 6 \cdot 110 = 660 and 14 \cdot 110 = 1540. But 6 and 14 are not
relatively prime, which one would hope would be the case for a gcd algorithm. Also,
not every number in the range (110,30) is an integral factor of a number in (1510,10)
and in (700,10)—an example is 110 itself. Moreover, there are several numbers which
would appear to be better than 110, which don't even appear in the iterations of the
algorithm: 6 \cdot 116 = 696 and 13 \cdot 116 = 1508, 7 \cdot 100 = 700 and 15 \cdot 100 = 1500,
8 \cdot 88 = 704 and 17 \cdot 88 = 1496, and so on. Presumably the reason that these problems
didn't affect the performance of Sreenivas' algorithm is that he usually only calls his quasi-gcd algorithm on successive harmonics.

The problems with Sreenivas' method can be eliminated by using a different definition of divisibility (keeping all the other definitions):

\[ (G, g) \mid (A, a) \text{ iff there exists an integer } k \text{ such that } A - a < kG - kg < kG + kg < A + a. \] (2.8)

This definition is considerably more plausible than the other; it says that \((G, g)\) divides \((A, a)\) if every number in \((G, g)\) divides every number in \((A, a)\) (with the same multiplier). We now come to the problem, however, that we can no longer use the Euclidean algorithm: \(G \mid X\) implies \(G \mid kX\) (and \(G \mid -X\)), \(G \mid X\) and \(G \mid Y\) imply \(G \mid (X \pm Y)\), and \(G \mid X\) and \(G \mid (X + Y)\) imply \(G \mid Y\), but it is not true that \(G \mid X\) and \(G \mid (X - Y)\) imply \(G \mid Y\). Take, for example, \(G = (100, 5)\), \(X = (200, 10)\), and \(Y = (90, 5)\). We thus need to find an alternative to the Euclidean algorithm. This is not hard: the gcd's of \((A, a)\) and \((B, b)\) are all of the non-null intersections of \((A/n, a/n)\) and \((B/m, b/m)\), for \(n\) and \(m\) being arbitrary relatively prime integers. The intersection of \((X, x)\) and \((Y, y)\) is defined as \(((C + D)/2, (C - D)/2)\), where \(C = \min(A + a, B + b)\) and \(D = \max(A - a, B - b)\). The problems with Sreenivas' definition do not occur with this one. Implementing it is quite easy, as well: simply take all relatively prime pairs \((n, m)\) and see if there is any intersection, for \(n\) and \(m\) small enough that \(A/n\) and \(B/m\) are not too small to be considered (there are an infinite number of gcd's, so we can't find them all). It is straightforward to write an algorithm for doing this which is order \(\min(N,M)\), rather than order \(NM\), where \(N\) and \(M\) are the maximum values of \(n\) and \(m\), respectively. The resulting gcd operation is associative (Sreenivas' is not); if one wants to find the gcd's of \((A, a)\), \((B, b)\), and \((C, c)\), first find all the gcd's of \((A, a)\) and \((B, b)\), then collect all the gcd's between each of those gcd's and \((C, c)\). When one is using this to find the gcd of harmonic locations, and there happens to be several gcd's with means (the first element of the pairs) greater than a threshold (say 60), the one with the greatest standard deviation (the second element) is taken.

To improve the accuracy of the result, instead of just returning the mean of the quasi-gcd of the harmonic locations, that mean is used to compute a least cost estimate of the fundamental ([39]). If one assumes that the \((X_k, x_k)\) used in the quasi-gcd computation represent independent Gaussian probability distributions with means \(X_k\) and standard deviations \(x_k\), specifying the probability that the measured harmonic is in a particular place, then the optical estimate of the fundamental is:

\[ F = \frac{\sum_{k=1}^{N} w_k X_k n_k}{\sum_{k=1}^{N} w_k}, \] (2.9)

where \(N\) is the number of harmonics, \(n_k\) is the integral multiple of the fundamental for the \(k\)th harmonic (found from the quasi-gcd), and \(w_k = (n_k / x_k)^2\).

---

9The answer is not unique, as we saw for 700 and 1510.
Figure 2.16: The top graph shows the speech signal that has been used in deriving the preceding figures as well: male speaker KS saying "Variability is a bugbear of those who use version eight." Below that is the pitch contour obtained from the mathematical pitch tracker, and on the bottom is the f0 contour derived from the glottal closure locations, as determined from the EGG signal. The time axis is 3 seconds long, and the range of the pitch contour displays is 0 to 180 Hz. The reason that the computed pitch contour extends beyond the voiced regions (indicated by the f0 contour from the EGG) is that a fairly wide analysis window is used (a 64 msec Hamming window), and some periods are included in the tails of the window even when it is centered on a voiceless region. This is also why the computed contour is so smooth.

It is something of an open question as to what the $x_k$ should be, since the harmonic finder only determines the $X_k$. It is important to distinguish several quantities here: the bandwidth of the spectral peak, the deviation of the (true) harmonic component from being a multiple of the quasi-fundamental, the measurement error in the process of finding the harmonic, the probability that the peak found is spurious, and the resolution of the human ear. For the $x_k$, the only ones of interest are the second and the third. Currently the $x_k$ are set to $0.15X_i$ Hz, since it was expected that aperiodicity would shift the harmonics proportionally to their frequencies. The true behavior of harmonics in the spectral representation of quasi-periodic signals still remains to be investigated. The validity of the independence assumption in the estimation of the fundamental should also be studied.

An example of the output of the algorithm is shown in Figure 2.16. With very little development and not much testing, the pitch tracker gets about 85% of all voiced analysis frames correct (i.e., 15% gross errors, using EGG f0 contours as the standard);
this no doubt could be improved upon considerably by further work. It is currently unuseable as a voicing detector. The mistakes were almost entirely caused by the harmonic finder labeling a spurious peak as a correct one. Unlike Terhardt’s and Duifhuis’s methods, this one is very sensitive to the presence of a spurious peak, since it takes the quasi-gcd of all the harmonics. This failing could perhaps be alleviated by taking the quasi-gcd of several subsets of the harmonic locations; this could at least protect against a single bad harmonic.
2.3 Summary and Conclusion

In the first chapter, I attempted to clarify and expand upon a variety of issues associated with pitch and pitch determination. The centerpiece of the chapter was the typology of pitch definitions, which used the categories of "production pitch," "mathematical pitch," and "reception pitch." This typology was used as a framework from which to consider how voicing, quasi-periodicity, and \( f_0 \) could be defined. I also developed the consequences which the typology has for the inputs and outputs of pitch trackers, and on the areas in which particular pitch trackers should be applied. After discussing the distinctions between theories and implementations, and between the scientific and engineering reasons for implementation, I showed what consequences those distinctions have for interpreting flaws of pitch trackers, and for how pitch trackers should be tested.

In this chapter, I have presented two new pitch determination algorithms: one which attempts to determine glottal closure locations automatically from a speech signal, and another which attempts to find the quasi-fundamental from a quasi-periodic signal. The first of these was based on a peak-finding algorithm applied to a processed LPC residual, augmented by an extrapolation procedure to handle cases where the LPC residual failed. This was tested on an 80 sentence corpus of speech and corresponding electroglottograph signals, and was shown to be very accurate and fairly reliable. The second of these was based on a quasi-gcd algorithm applied to the harmonic locations found from a short-term spectral representation. The results of limited testing of this algorithm were promising.

Part of the motivation for the material in the first chapter was to attempt to justify the content of the second, and to distinguish the route taken here from the other myriad directions that have been taken in attempts at solving the problem of determining the pitch of speech signals. The major decisions entailed by the algorithm for determining production pitch, described in section 2.1, were stated explicitly and argued for in the first chapter:

- Mathematical pitch is comparatively not as useful or as informative as production pitch.

- Currently it would appear that psychoacoustic pitch, while extremely interesting in its own right, is also not as useful as production pitch, and it is now much too premature to try to implement a general psychoacoustic pitch tracker anyway.

- Production pitch should be represented in terms of entities meaningful to the production process, not in terms of structural waveform primitives in the speech signal.

- A production pitch tracker is not obliged to be able to handle arbitrary non-speech inputs.
• Speed, cost, memory requirements, and related evaluation metrics are of a qualitatively different sort from accuracy.

• A production pitch tracker should be tested with a glottographic standard.

Together, these statements determine an ambitious project, one which was only begun here. This is best regarded as the first step on what I would regard as the proper path for pitch determination to take, certainly not the last. For the production pitch tracker, more work is needed on the basic transformation, to be able to better handle voiced fricatives and sounds with little high frequency energy. Improvements are also needed in the peak finding algorithm, to be able to distinguish extremely irregular glottal closures from noise; this might be done with a better representation of context constraints. The extrapolation stage is a source of inaccuracy and should ultimately be eliminated. In general, it should be possible to greatly improve the algorithm's accuracy and dependability. For this to be possible, it will also be necessary to use a standard with better accuracy and dependability; perhaps a glottograph more reliable than the EGG should be considered. Certainly a more sophisticated method than the one used here should used for determining the glottal closure times from the glottographic signal.

For the mathematical pitch tracker, a more reliable procedure for recognizing harmonics should be used, and some method needs to be found for making the quasi-gcd operation more robust against spurious harmonics. On the scientific side, the definition of quasi-periodicity needs to be more thoroughly investigated, and the behavior of harmonics in a time-frequency representation of a time-varying signal should be studied more.
Appendix A

The Structure of a Pitch Tracker

The purpose of this appendix is to present a general view of the process of pitch determination, following the somewhat pseudo-philosophical orientation of the first chapter. It is not immediately evident that the ideas here actually provide much substantive insight or lead to effective approaches to pitch detection, so this has been relegated to the status of an appendix.

It is no less difficult—no less futile—to attempt to come up with a general structure for all pitch detection schemes, than it is to attempt to find a general structure for all novels. Beyond tautological statements that all of them have a beginning, a middle, and an end, there is extremely little that holds universally. However, there are aspects of the two problems of this chapter which they share, and which would benefit from a somewhat abstract discussion covering both of them. Usually descriptions of pitch determination algorithms are split thematically and chronologically into preprocessing, processing, and postprocessing stages. I would prefer to organize my discussion under the rubrics of primitive extraction, local processing, and context constraints and control structure. This is mostly for exposition; the algorithms themselves are by no means separable into such neat modules. Sometimes a stage is trivial or nonexistent, it is often hard to assign a particular part of the computation to one of them, and the “stages” sometimes do not occur in order but instead act together. But the issues and ideas in these areas are fairly separable and interchangeable, and they apply broadly to many problems other than in pitch determination; it is helpful to consider them with this in mind.

A.1 Primitive Extraction

Both of the algorithms in the second chapter (unlike, for example, correlation) involve discrete objects in their computation: glottal closures, and harmonics. It is the responsibility of the primitive extractor to obtain from the input signal the
basic objects which local processing will manipulate: hypothetical glottal closures and hypothetical harmonics. Generally this works blindly, locally, and quickly, and yet despite that attempts to as much as possible remove the "bad" and/or accentuate the "good." Note that we aren't making an assumption of ignorance which might hurt the algorithm; if there is too much knowledge in the extraction stage, then it might as well be called local processing, and a trivial primitive extraction has already been done. As pointed out above, there is no criterion providing a sharp dividing line between these two stages.

By and large, an algorithm can only be as good as its primitives are. There are three kinds of errors the extractor can make: it can accept a spurious primitive, it can reject a true primitive, and it can give a noisy measurement/estimate of an attribute of a primitive, such as its time or frequency. Now for all of these errors, it is conceivable that by considering all the hypothetical primitives returned by the extractor, it would be possible to determine that some of them don't belong, that something is missing, or that something is mismeasured, by reasoning from some deviance in the common pattern. However, there is an important difference among these errors: when a spurious primitive is accepted, the local processing need only reject it, whereas if a primitive is missing or measured incorrectly, it is necessary to go back to the original signal and try to straighten things out, a much more thankless task. Also, it is impossible to detect a mistake by the primitive extractor when there is no difference in its output from the case of correctly processed, unusual input. This would appear, just on a priori grounds, to be a more serious problem for missing and mismeasured primitives. Moreover, it would seem that the method whereby correction is done will necessarily have to incorporate a lot of idiosyncratic knowledge about the extraction algorithm, in order to pinpoint the source of the difficulty. Thus effort to develop that part of an algorithm will be almost entirely wasted if the primitive extractor is changed significantly.

It is for these reasons that in both the algorithms presented here, no effort is made to second-guess the primitive extraction, except to remove spurious primitives. If primitives are missed, that might well cause the algorithms to fail, and if there is inaccuracy in the primitives, there will be inaccuracy in the final result as well. For mathematical pitch (finding the quasi-fundamental), this is not a crucial issue. Often even when some harmonics are missing, the signal corresponding to the remaining harmonics has the same quasi-fundamental as the original one. Actually, dealing with spurious harmonics is more of a problem, so that the harmonic extractor might even be designed to reject a lot of valid peaks, so as not to accept too many spurious ones. But still, no effort is made to recover lost primitives, while some attempt is made to be robust against spurious ones. For production pitch (finding glottal closures), it is absolutely vital that none be missed, because it is difficult to recover from such a mistake.
A.2 Local Processing

There is of course more to a pitch tracker than just a primitive extractor. However, there are two very different reasons for this incompleteness in the problems of finding the quasi-fundamental and of determining glottal closures. The reason we aren’t done with the first is that the primitives are not the same sort of thing as the desired output; the harmonics have to be combined in a quasi-gcd algorithm to calculate the quasi-fundamental. This is an example of local processing. Context constraints play little or no part in the algorithm; it is finished with the local processing. Particularly since the definition of quasi-periodicity admits signals such as phase modulated square waves, it is hard to see that any kind of continuity constraints are justified. Those that are justified, are too loose to be of much help. The algorithm for determining glottal closures is just the opposite. Here the primitives are the same sort of thing as the output, and local processing is irrelevant. If the primitive extractor were perfect, the algorithm would be finished; since of course this is rarely the case, context constraints are necessary to eliminate the bad primitives.

One minor issue which the two do share is in how the certainty of the primitive extractor is interpreted by the later processing. Note that in the definitions behind both algorithms, there is no notion of a degree or scale; something is either a glottal closure (harmonic) or it isn’t, and the relative strength or weakness of the closures (harmonics) in the signal has no effect on the corresponding pitch. Still, the algorithm needs to know for each primitive how confident the extractor is that it isn’t spurious. For both closures and harmonics, a monotonic function of amplitude is the most obvious thing to use, but there are equally obvious problems with the suggestion: sometimes spurious primitives have quite large amplitude, and valid small amplitude primitives may be no harder to detect than large amplitude ones.

A.3 Context Constraints

A pitch tracker could be considered to be a function from an input signal to a characterization of the pitch of that signal. For a particular definition of pitch, if one were to consider this mapping from the infinite number of possible valid inputs to their correct outputs, it would no doubt be discovered that the possible responses to a particular small region of an input is also conditioned upon what is going on in the surrounding area (one needn’t appeal to probability). When a pitch determination algorithm has trouble making error-free local decisions, it thus might benefit from using non-local knowledge to restrict and score the possibilities. While this is most directly applicable to detecting glottal closures, it can also be useful in determining mathematical pitch—despite my past and future statements about the aesthetic offensiveness of the practice—when the domain of input signals is restricted and the signals are noisy.

Complete knowledge of the constraints (if that makes sense) is not necessary; in fact, none may be needed at all; it is certainly conceivable that in some cases local
information is sufficient, even when there are context constraints that might be used. Moreover, even complete knowledge of the constraints and the context in a particular case is not enough to predict exactly what happens locally; some local information is always needed to determine what the innovation is. Because of this, constraints are always a little wrong: they predict several things as being possible when in fact only one of them occurs.

Since the use of constraints (necessarily) follows primitive extraction, constraints between primitives are what are most usable in an algorithm. One might consider basing the constraints entirely on the primitives, by studying the mapping from the extracted primitives to the result. Instead of working with the regularities in quasi-fundamental functions and glottal closure sequences, regularities in the estimated, quantized quasi-fundamentals and closures would be used. But there are problems with depending on constraints between primitives rather than constraints "in" the data. Surely no matter what kind of primitive is chosen, there will be some continuity constraints or some other context effects which can be found, but these constraints will arise from the basic constraints in the signals, masked by the noise in the measurements of the primitives. Not only does this mean that it will be difficult to make an accurate estimate of what the constraints are, but if the primitive extractor is altered, or if a different class of input signals is used, the constraints are likely to cause disaster. If, for example, waveform zero crossings are used as glottal closure primitives, or if the spectrum is sampled coarsely when finding harmonics, the noise in the primitives is likely to overwhelm the underlying constraints. Also, effort spent on learning about the eccentricities of a particular primitive extraction scheme is wasted when a better one comes along, whereas constraints based on the properties of speech signals (or whatever the input class is) will always be useful. Of course, noise in the primitive extraction can't be ignored; it must be considered when the constraints are being used. But the issue should be kept separate, and the constraints should originate only from the input signals.

Analogous statements can be made about how the constraints are represented in the algorithm; just as primitive extraction approximates the information that the constraints apply to, the control structure of the algorithm causes an approximation to the constraints themselves. For example, since the control structure of most past pitch tracking algorithms has been to move straight through the signal, making irrevocable decisions, it is quite natural that the way constraints are most often approximated is by a Markov model. As with primitives, no matter what model is chosen, the underlying constraints will have some projection onto it. The more appropriate the model and primitives are, the more effective and accurate the constraints used will be.

One way of thinking about constraints is that they help the pitch tracking algorithm to solve an exponential search/optimization problem. Suppose that a tracker of mathematical pitch finds \( N \) (around 100-200) quasi-fundamentals per second, and that each quasi-fundamental can take on \( L \) (around 100) different values. Then there are \( L^N \) different one second long pitch contours that the pitch tracker can conceivably return. In finding glottal closures, a pitch tracker has to decide \( M \) (8000-20000) times a
second whether that sample is the moment of a glottal closure or not: there are thus $2^M$ different one second long outputs that it can return. In this view, the problem of pitch detection is to determine the best $N$ ($M$) long path through a graph, corresponding to the input signal. The way this is done is by suitably combining local information, which is derived from the signal, with global information about paths, which is universal. This makes intuitive sense: when we visually mark pitch periods in a speech signal, we combine our local ability to recognize periods in isolation, with the knowledge that they generally appear at regular intervals. In finding a quasi-fundamental contour, we combine local evidence about what is a likely quasi-fundamental, with information about what the quasi-fundamentals are nearby. A Markov model is a simple-minded way to implement this balance of local and global, which makes clear how the problem can be viewed as a search for the best path through a graph; node costs are determined by the local, signal-dependent information, and arc costs (transition probabilities) are determined by the global, fixed information. But all the other models we will look at in the next section also reflect this trade-off between agreement with what the local data would indicate, and agreement with an a priori assumption about global properties.

Before moving on to a discussion of some control structures that might be used, I want to first point out some problems with the alternatives to combining local and global information.

A concomitant danger of using context constraints is that the algorithm will get in trouble when the assumptions of the constraints are violated by the input. One might suggest that the way to avoid this problem is to use no context constraints (global information) at all, but just depend on local information. However, it should be recognized that there is something of an uncertainty principle, an unavoidable tradeoff, at work here: the tighter and more limiting one’s constraints are, the faster and more accurate the performance will be on “valid” input, but in turn, it will be more difficult to recover from “invalid” input. Having no constraints is a model just as any other, albeit a trivial one: it allows crazy transitions; all paths are equally likely. It might happen to be better than more restricting constraints (as in the case of mathematical pitch), or it might be worse, but it isn’t different in a way which makes it exempt from this general tradeoff. In using a deterministic, no memory algorithm, there won’t be any problems caused by a bad decision ruining later decisions, but it won’t be able to take advantage of any help that nearby decisions might be able give, either. Regularities in the input space might very well drastically reduce the possibilities for the output space—it might even be brought down to less than exponential size. To take advantage of these regularities, we have to make some assumptions; as with any optimization technique that entails not looking at some paths, it is always possible that the best path will be missed, if those assumptions are not correct for the given input. But by not using the regularities, the algorithm might be prone to returning incorrect outputs which would have been ruled out by those constraints. It might be slower as well, because it won’t be using the constraints to cut down on its search space.

The opposite extreme is to use context constraints, but to almost completely ignore any local information. This is actually the way context constraints are most often used
in pitch trackers: "postprocessing" amounts to exactly this kind of approach. Here, context constraints are not used until after everything has been done, and all that is left is a pitch contour which needs correcting. Since a linear smoothing filter is out of the question, nonlinear smoothing is done, for example by convolving with a median filter ([88]), or by using dynamic programming to determine which values are unbelievable and should be replaced ([69]). This is clearly the worst way to use context, and should be considered a last resort. It has to rely entirely on a characterization of what acceptable pitch contours look like, since it has no access to the original signal or to the alternative choices that were available when the pitch contour was being produced; it could have used this other information to strengthen the constraints, and to supply correct answers instead of interpolating from the neighbors. All of the control structures below could be reconsidered as methods of postprocessing: they don't have to incorporate local information, though they can. But obviously, if there is extra knowledge about pitch in the postprocessor, it should be used when the decisions are being made, not after all the data for those decisions (local information) has been thrown away.

A.4 Approximating Constraints: Control Structures

In this section I will give a short overview of five different control structures, each of which might conceivably be used for a pitch detection algorithm: Markov models, full-fledged dynamic programming, island-driving, relaxation, and parsing. The kinds of optimization problems that these algorithms attempt to solve are widespread; much of this overview is pertinent to applications in formant tracking, image processing, graph search problems, and data smoothing, just to name a few.

This is not intended to be a tutorial; readers unfamiliar with this material will no doubt find my descriptions much too terse and superficial, and will have to look to the references cited for a simpler and more thorough introduction to these ideas. Instead, my intent, for each approach, is to give a brief, abstract description of it, to show its similarities and differences with respect to the others, to suggest how it might be applied to pitch tracking, and to list some of its essential advantages and disadvantages.

Throughout this discussion, it would be beneficial to keep the following four principles (or issues) in mind:

- One way to make an a priori evaluation of a pitch determination algorithm is to consider how well the chosen primitives, constraints, and control structure would work if the processing were reversed and they were used for synthesis or generation. The more like actual, typical input signals the results would be, the better the algorithm.

- The source of context constraints is in particular facts about how the signal is produced. Therefore, one might expect that the most effective way to represent
constraints and to process signals, would be in a manner akin to the way it actually occurs. (This is another way to see that there are no context constraints universally appropriate for mathematical pitch, since it has no source.)

- In the same way that a distinction should be made between issues of pitch definition and issues of pitch detector implementation, a sharp separation should be maintained between the projects of determining what the constraints are in the signal, and of deciding on how to approximate those constraints. Implementation of constraints (via a control structure) is independent of finding them; one could at least imagine a pitch tracker which simply considered every member of the output space and evaluated it according to the constraints, i.e., a pitch tracker with no approximations of the constraints.

- The mere fact of convenience or elegance of a control structure is not enough to recommend it; this is a very special problem and it might turn out that the best way to solve it involves a seemingly "ad hoc" way of using constraints.

### A.4.1 Markov Models

Perhaps the most obvious way to invoke context in a pitch detector is to assume that a pitch value has a particular probability distribution conditional upon the previous value. More formally, one assumes that pitch contours are generated by a first-order Markov source: if \( X_1, X_2, \ldots \) is a sequence of random variables corresponding to the sequence of pitch values, then \( p(x_n|x_{n-1}) = p(x_n;x_{n-1}, x_{n-2}, \ldots) \). Markov models have been used in many areas; to my knowledge Baker ([12]) was the first to suggest its use on pitch tracking. Secrest and Doddington’s pitch determination work ([99], [100]) is an example of explicit use of this theory, but many pitch detectors implicitly do this as well, by using the previous estimate of pitch to help in the next estimate.

As hinted earlier, the way that this is used in pitch tracking is to first find a sequence of crude probability distributions \( p(X_i = x_i) \). This comes from the local processing: correlation or AMDF (ignoring for the moment their problems) might be used to obtain a score for each of several possible pitches for each analysis frame. This is combined with a fixed transition probability function \( c(X_{i-1} = x_{i-1}, X_i = x_i) \): the task of pitch tracking is to find the path \( x_1, x_2, \ldots \) optimizing the likelihood \( \prod c(x_{i-1}, x_i)p(x_i) \). The way this is done is with the Viterbi algorithm ([126]). Suppose that there are \( L_{n-1} \) different possible pitches \( x_{n-1} \) for \( X_{n-1} \), and the algorithm has the \( L_{n-1} \) single best paths leading up to each of them. Then, because of the Markov assumption, the best \( L_n \) paths leading up to each of the \( L_n \) possible pitches \( x_n \) of \( X_n \), can be found by determining which of the \( L_{n-1} \) paths maximizes \( C(x_{n-1})c(x_{n-1}, x_n) \), where \( C(x_{n-1}) \) is the likelihood of each of the best partial paths leading to the \( x_{n-1} \). The best path up to each \( x_n \) is a continuation of the path to the \( x_{n-1} \) found in that maximization procedure. Thus an \( O(L_{n-1}L_n) \) operation has to be done for each \( n \); the single best
pitch contour is found by choosing the path with the greatest likelihood from among the $L_N$ paths leading to the last Nth frame.

This algorithm was just described in a way appropriate to a mathematical pitch tracker. The control structure for production pitch (glottal closures) is analogous: several partial paths are maintained at once; they are extended by adding new glottal closures on to them in a way which maximizes the product of the path likelihoods and inter-closure time transition likelihoods. The local measure of goodness might be the height of the peaks in the output of the primitive extractor. It is a little more complicated than the other, because the paths are not of equal lengths and the processing is not frame by frame (primitives are skipped), but the principle is the same.

This is fast and simple, and it does improve performance. As was mentioned earlier, perfect characterization of constraints as implemented in control structure is first of all impossible and second of all unnecessary. However, it is important to recognize the inadequacies and assumptions of the model. All of the problems with it can be seen as a consequence of the observation that pitch contours just don't look like typical sample functions from a Markov process. "Random Walk" models, which Markov models are instances of, are more accurate for stock indices than pitch contours. Pitch contours for speech typically have broad peaks and valleys, with a "micromelody" of segmental effects superimposed, and interruptions caused by voicelessness. This means that the algorithm will consider as a possibility a pitch contour which in fact is not possible, so that the algorithm will be slower and more error-prone than necessary. Not only is the model too weak, but it is too strong: it will rule out some pitch contours because of bad transitions even though in the larger context, those transitions are perfectly acceptable and even predictable.

A.4.2 Full-Fledged Dynamic Programming

The Viterbi algorithm is just a special case of dynamic programming, which predates it ([14], [51]). Dynamic programming is more of an approach than a specific algorithm; this is true of the control structures of the next few sections as well. It has been applied to a vast variety of problems, including ones in operations research and in speech recognition; see, for example, [15], [17] and [97]. The unifying idea of the approach is to reduce a search through an $O(e^n)$ space to an $O(n)$, $O(n^2)$, or $O(n^3)$ problem, by using a recursive optimization method. This is dependent upon the ability to formulate the problem in such a way that at each stage decisions can be made and "state" information constructed solely on the basis of the state of the previous stage. One case of this is context-free parsing (to be discussed later); another is the Viterbi algorithm.

I am treating this in a separate section from Markov models so as to be able to present the simplest case first, not because I think there is any important difference. It is true that there are well-known methods for statistically estimating/training the
parameters (the transition matrix) of an unknown Markov model from data generated by it, and this is not the case for all dynamic programming algorithms, but for us this is a small sacrifice to make to obtain the added generality.\footnote{Actually, in the pitch detection field it is currently no sacrifice at all, because so far as I know no one has ever taken advantage of the formal techniques for estimating the parameters; instead people have just improvised rules based on their intuitions.}

In the Viterbi algorithm, it is entirely unnecessary to restrict oneself to just a computation on \( x_{n-1} \) and \( x_n \); a structurally identical procedure could do any computation that involves only whatever information is being maintained at each stage. This is not simply a matter of going to a multiple order Markov model, i.e., using \( p(x_n|x_{n-1}, \ldots, x_{n-k}) \), for \( k > 1 \): that is not the way to take larger context into account. This is also a slightly more general idea than moving to hidden Markov models. One could, for example, keep track of not only the last pitch value for the partial paths at each stage, but also an estimate of the current slope. Or, when \( x_n \) and one of the \( L_{n-1} \) partial paths are being tested for compatibility, a least cost parabola could be fit to the last six values of the path, and extrapolated to compare to \( x_n \).

One reason this is appealing for production pitch is that it would seem to be analogous to the way the signal is produced: at any moment, the vocal source is in some physical mode or state; there is a certain amount of randomness so that if the system goes into that same state several times, it varies slightly in what it will do next (i.e., in what its next state will be). A person can be producing the same instantaneous \( f_0 \) value with many different states; knowing which of the states he is in is much more helpful than just knowing the current \( f_0 \). The fact that the processing moves deterministically left to right (or rather, forward in time), also means that information on the current state can inform the primitive extraction and local processing of the next frame.

\section*{A.4.3 Island-Driving}

"Island-Driving" techniques start out with the same data as the dynamic programming algorithms: a ranked list of candidates is given for each frame. Also like the other algorithms, it uses fixed knowledge of context constraints: in knows roughly what sequences of candidates are likely, possible, unlikely, and impossible. However, unlike the \( O(n) \), left-to-right Viterbi algorithm, it processes the frames in no fixed order. It proceeds by first finding a stable, unambiguous "island" frame, and then attempts to grow out from it, by finding a chain of candidates from the frames in that neighborhood. These chains are analogous to the partial paths of the previous section, except that they don’t start at the beginning of the data. This is done in other places as well, so that eventually chains start meeting, causing them to either be merged if they are consistent, or rejected, if they are not. A formalization of this idea, which describes the appropriate data structure and priority scheme for the chains, is found in
the $A^*$ algorithm; this is a heuristic graph-search algorithm which was given its cryptic name by Nils Nilsson ([45], [72]). Woods ([132]) demonstrated the applicability of $A^*$ to a problem in speech recognition: given a device which will return a ranked list of candidate words covering an arbitrary quantized time range $(t_1, t_2)$, and a grammar which determines acceptable word sequences, find the optimal word sequence for the sentence in $(0, T)$. No one has used this theoretical framework in pitch determination yet, but there have been a few cases of using operations akin to island-driving, including some unpublished work by Barbara Caspers at Bell Labs on pitch tracking from multi-pulse excitations, Specker's "post-processing" method ([110]), and the production pitch tracker described here. The method is equally usable for frame-by-frame mathematical pitch tracking and for eliminating spurious glottal closure labels.

For this algorithm to be usable, the context constraints must be expressible in a form where some kind of compositionality holds: the cost for the result of merging two chains must be computable from the cost (and/or state info) of those chains. (Rejecting a merge amounts to giving the merge a very high cost; growing a chain amounts to merging with a singleton chain.) For the algorithm to be admissible, i.e., for it to be guaranteed to make no mistakes, there must be a heuristic evaluation measure for chains which satisfies a particular property ([72]). The heuristic is the function which makes an estimate of the cost of the best total path using a given chain. This usually takes the form of $f(x) = g(x) + h(x)$, where $f$ is the heuristic, $x$ is the chain, $g$ is the cost of the chain, and $h$ is the guessed cost for the best (not yet found) complete left and right continuations of $x$. The criterion for admissibility is that $h(x) \leq h^*(x)$, where $h^*$ is the correct cost for the best continuation. When $g(x) =$ length of $x$ and $h(x) = 0$, $A^*$ reduces to breadth-first search; this is admissible, but it also takes exponential time. Woods proposes two alternative heuristics: shortfall scoring, which is admissible but slow, and shortfall density scoring, which is not admissible but is faster.

This control structure has several advantages. For speech signals, it would seem to be just the right thing for dealing with the fact that the onsets and offsets of voicing, and mixed excitation, are always difficult to process; by growing outwards from the stable sonorant regions one might hope to be able to handle them more easily. It also allows a great deal of freedom of choice for the cost function, and it is potentially very robust. It does have its disadvantages, however. Depending of course on the cost function, the algorithm can get rather slow, and it is always complicated. Also, because (unlike the algorithm of the preceding section) it does not work deterministically through the data, one no longer has the nice feeling of processing the data in a way similar to that with which it was produced, it is more difficult to use the context constraints to inform the primitive extraction and local processing, and it is no longer possible to use a short fixed length buffer to fulfill all memory requirements.
A.4.4 Relaxation

Relaxation refers to a class of algorithms which if anything is even larger than that of the previous section. Its most direct applications, in the form of “constraint satisfaction” and “constraint propagation,” have been in various problems in vision, but it can also be seen in such varied areas as non-Van Neumann computation models (e.g. simulated annealing, Boltzmann Machines, and associative memory), finite element methods in physics, and non-linear data analysis (“greedy” filters such as the median filter) ([48], [52], [43], [94], [21], [44], [22], [127], [82]). As with the previous control structures, the input comes in the form of a sequence of ranked lists of candidate labels for successive times (or nodes, or image locations). Unlike the previous algorithms, however, this one does not maintain any auxiliary data structure for keeping track of tentative interpretations of the input. Instead, it essentially changes the input itself, by updating the weights on each of the labels of the nodes, until the interpretation corresponding to taking all the best local labels satisfies the given constraints. In a single update operation on a node, each possible label on that node is considered in relation to the labels (and their weights) of nearby nodes. If a label fits in well with labels on neighbor nodes which have high weights, then its weight is increased; otherwise it is decreased. The specific update formula used varies considerably across applications. The procedure which determines the order that nodes are updated also varies; two ways that are often used are (1) update every node in the graph, sequentially, following the connections of the graph, then repeat, and (2) randomly select a node, update it, and repeat.

This algorithm requires that the constraints be describable in terms of an evaluation of a local configuration of labels; there is no opportunity here, as with the previous cases, to have constraints which concern themselves with large regions of the data. However, this may not be a drawback in some cases, because one of the potential advantages of the algorithm is that the local constraints can “propagate”: on each update iteration, nodes which were previously affected by some neighbors will in turn affect other neighbors, so one might hope that eventually apparently global constraints could be satisfied. When this formulation of the constraints can be accomplished, this algorithm has a conceivable advantage over an algorithm which uses global constraints explicitly: it isn’t necessary to specify a particular region size on which to use the constraints; when the data is well-behaved, hardly any propagation will be necessary, and the answer will be arrived at quickly, and when the data is noisy or ambiguous, the constraints will propagate however far is necessary to figure things out. Of course in many cases the obvious way to express the constraints is locally, so there is no choice to be made.

Relaxation is also different from the previous algorithms in that it is not as easily applicable to the problem of removing spurious glottal pulses, as it is to the problem of improving decisions in mathematical pitch tracking. In the glottal pulse case, the neighbors of nodes change whenever one is deleted or reinserted, and this makes implementation problematical. There have been some attempts at “hierarchical relaxation”
they usually turn out to be a hybrid between relaxation and parsing.

There are several serious problems with relaxation techniques. These are certainly not enough to eliminate relaxation from a list of useful techniques; it has been applied quite successfully to many problems, and in some of the cases where it hasn’t worked very well, there is still no superior alternative. The first, obviously, is that the constraints have to be expressible in a particular form, and the approximations necessary to put them in this form may be so egregious that it isn’t worthwhile. Secondly, despite the conceptual simplicity of relaxation methods, they typically end up being slow. Often the update procedure is expensive, and it must be done on every label of every node, several times. Thirdly, no one has yet found a good termination criterion. As mentioned above, the algorithm has no state beyond the changing label weights; unlike the other algorithms discussed, at any moment the procedure is in a state which is structurally identical to the one it started with. As one would expect, this means that it is hard to fix upon a way to stop “relaxing.” For the very simplest kinds of relaxation, the algorithm can (in fact must) be terminated when it reaches a fixed point in its multi-dimensional state space, but that doesn’t work for the more complicated versions. This problem is aggravated by the fact that for some kinds of relaxation, the longer that the algorithm operates the more it forgets about the original input; if it isn’t stopped soon enough, it will return an analysis which bears little relation to the input data. Fourthly, the algorithm deals with ambiguity inappropriately. When there are two reasonable interpretations of the input, relaxation almost arbitrarily selects one of them and emphasizes it, to the detriment of the other one. It is impossible, without serious modifications which have yet to be made, for the more complicated relaxation algorithms to entertain more than one hypothesis at a time (this is possible in Waltz’s simple case [127]). Unlike most other algorithms, there are no second place winners in relaxation; it cannot return a ranked list of interpretations, or go back and find another answer if it is “told” that the answer it gave was wrong. Fifthly, we currently have only a limited understanding of the behavior of relaxation techniques; it is thus difficult to predict or explain how it handles a given input, or to intelligently adjust parameters.

### A.4.5 Parsing

When the distinction between different labels is qualitative rather than quantitative, the problem of finding the best label assignment can be reconsidered as an instance of a parsing problem: given the symbols and their weights corresponding to each time (node), and a grammar for the language which includes all acceptable strings of symbols, find the most likely parse (the list of rule applications which will generate the best labeling). This is sometimes used in the analysis of waveforms and two dimensional curves ([82], [115], [24]): the primitive extractor returns things like “long plateau” and “sharp upslope,” and the parser determines from this where the peaks are, using a grammar which describes in general how peaks can be made up of
those primitives. Languages (the sets of possible symbol strings) vary in complexity; some example classes of languages are finite-state, LR(k), and context-free. There are standard methods of parsing most of these kinds of languages ([2], [41]); when parsing is used in applications such as the one we are considering, these algorithms are often generalized in a relatively straightforward way to do the following:

- Allow several possible symbols (primitives) for a single location, which the parser has to choose between.
- Allow weights to be given to the primitives.
- Allow costs or probabilities for the different rules in the grammar.
- Allow attributes to be associated with the primitives (things like height or slope); these attributes can be used by the grammar rules to compute weights and attributes for the nonterminals.
- Allow error-correction (handle insertions and deletions).

K.S. Fu's book ([35]) is an excellent introduction to this approach.

To take an example, suppose that the primitive extractor finds peaks in the waveform; these peaks could either correspond to the first peak in a pitch period (closest to a glottal closure), a subpeak in a pitch period, or a random peak in noise. There is only one symbol in this grammar, the peak; recognition (determining whether the input string is in the language) is trivial—all strings are acceptable, and in many different ways. The job of the parser is to find the best parse for the input, using attributes associated with the peaks such as their height, width, symmetry, etc. It should be clear that this could be redone with more primitives (big-peak, broad-peak, etc.), a bigger grammar, and less processing of attributes.

Grammar:
1. $S \rightarrow UT$
2. $T \rightarrow (\text{nothing})$
3. $T \rightarrow VUT$
4. $V \rightarrow P$
5. $V \rightarrow PV$
6. $P \rightarrow (\text{peak})$
7. $P \rightarrow (\text{peak})P$
8. $U \rightarrow (\text{peak})$
9. $U \rightarrow (\text{peak})U$

Comments:
1-3. Speech is made of alternating voiced and unvoiced parts.
4-5. Voiced speech is made of one or more periods.
6-7. Periods are made up of one or more peaks.
8-9. Unvoiced speech is made up of a succession of spurious peaks.
Rule Heuristics:
3. Voiced and unvoiced regions should be reasonably long.
5. Successive pitch periods should have roughly equal sizes (energy, amplitude), shapes, and lengths.
7. Successive peaks in a pitch period should be evenly spaced and should decrease in height.
9. Unvoiced peaks should be chaotic and narrow.

The “rule heuristics” are performed whenever the parser matches the right hand side of a rule, to give a weight to the nonterminal on the left hand side which is created. This is a very simplistic example, but it shows the essential aspects of the method.

The main advantages of parsing are that grammars are potentially very perspicuous and they can be easily changed. The main disadvantage is that too many sacrifices might be necessary to be able to use it, both in the primitive extraction and in the constraints. Most problems can somehow be reformulated in terms of production rules or finite-state automata, but a great deal might be lost in the process. Consider the use of parsing on speech waveforms. The use of structural primitives—things like zero crossings, valleys, etc.—is a natural accompaniment to syntactic pattern recognition (what we have been discussing). But: (1) none of those primitives is dependable as a potential location for glottal closures, so by restricting the domain of possible locations, an unnecessary amount of measurement noise is introduced; (2) structural primitives are totally inappropriate for speech signals anyway; for example, it is nearly impossible to express the notion of “sum of damped sinusoids” in those terms; and (3) it either necessitates binary decisions (the primitive is there or it isn’t), or it ends up making a crude quantization of continuous variation (short medium long); the use of attributes helps a little, and all algorithms have to eventually make the leap from continuous to discrete, but this algorithm forces it at a very early stage. If one is going to use structural primitives in the analysis of waveforms, then parsing is certainly one good way to combine those primitives: syntactic pattern recognition has been applied, with some success, to carotid pulse waves ([115]) and to EEG data ([37]); the problems with using it on speech signals evidently do not apply with the same force for those simpler signals.

Parsing also imposes compromises in how context constraints are used. Often the grammars indicate a hierarchical structure in the data that isn’t really there, but has been brought in so that it is possible to take advantage of the power of parsing algorithms. Thus in the case above, the concept of a “chain of pitch periods,” which is probably what one would want to apply constraints to, doesn’t appear overtly anywhere in the algorithm. For problems with a totally flat structure, such as mathematical pitch tracking, there is little that parsing can help with. Moreover, even when a tree structure can been imposed, still the user is restricted to associating his constraints with the application of a rule. Also, parsers can only be guaranteed to return the best parse if compositionality holds: in the rule A→BC, the best match to A must
come from the best match to B together with the best match to C. It is possible get around this problem without going so far as to save the exponential number of possible parses, but it is difficult.

Parsing is generally rather slow; the most commonly used context-free parsing algorithms are $O(n^3)$ in the input size. Yet it certainly seems as if problems such as pitch tracking should be $O(n)$. The algorithms expend a great deal of effort just symbol-pushing: manipulating the nonterminals such as $U$ and $T$ above. It often gets rather messy to either use LR($k$) parsing instead, or to make the context-free algorithm effectively $O(n)$, by cutting up the input data and pruning the parse tables. Even on absolutely unambiguous input data, the algorithm cannot easily break out of its set procedure and quickly return the right answer.

### A.5 Integrating Information

In this section I would like to touch upon two issues: (1) the use of more information than just the speech signal, and (2) combining pitch trackers.

Pitch detection is by all accounts very difficult; because there is also a big demand for its solution, one might consider trying to solve a slightly easier problem than determining the pitch from only the speech signal. Thus a phonetic transcription, information about background noise, information about the speaker’s pitch range (or speaking rate, or sex), and/or a corresponding laryngographic signal, to name just a few possibilities, might also be provided. It is important to keep widely separated two different ways of responding to this suggestion. The first is to say that this is a singularly fine place to apply “Artificial Intelligence Techniques.” Here the problem is generalized and rephrased so as to be asking such questions as the following instead: Is there a uniform but sufficiently general way to represent knowledge so that it is possible to incorporate a wide variety of different sources of information, and to add on new sources when desired? How should one reason from undependable, incomplete, uncertain, incorrect, and contradictory evidence? These kinds of questions are very much still open; it is not simply a matter of applying standard “AI Techniques” to take care of them. It is thus unrealistic to hope that this project will contribute any substantial improvement to pitch detection in the near future; on the contrary, it is the pitch detection problem that is helping AI research (by providing a useful testbed), not the other way around. This is roughly the approach taken by Dove et al. ([26]). The second response is to consider what additional information would be both useful and convenient to obtain, and then figure out how those particular sources of information should be combined. This is no longer AI; the way information is coordinated is likely to be ad hoc and incapable of being generalized. However, now integrating information does not represent a deep, difficult problem all by itself; it is fairly straightforward to find a way to use a particular, fixed source of information. Thus this approach is quite likely to bring about improvement in pitch determination, and in fact it has usually met with success when it has been tried (e.g. [123], [54]). As with the AI approach, there
will of course always be the unavoidable disadvantage that the resulting algorithm will less flexible and convenient because it requires more than just speech signals, but this is certainly a reasonable way to proceed when universality and convenience are not overriding issues.

It has occasionally been suggested that one route to solving the pitch detection problem is to combine the outputs of several pitch trackers ([46] pp.548-549, [19]). Each pitch tracking algorithm has its own specialty which would hopefully complement the others; together they might do better than any one of them would individually. When more than one tracker "takes responsibility" for a portion of the speech signal, some kind of majority vote or averaging would be performed with their outputs. Although no doubt some improvement could be obtained with this idea, it has limitations. I believe that in most of the places where good pitch trackers fail, all pitch trackers fail; clearly this approach will not help in those areas. Also, for the combination to work out, it is necessary to have good estimates of the time-varying, signal-dependent probability that each pitch tracker will fail, and it is likely to be necessary to assume that the failures of the pitch trackers are statistically independent of each other. Both of these requirements are problematic. Another problem is that because the outputs of different pitch trackers contribute to the final output, there is a danger that slightly different interpretations of "pitch" will be mixed into a meaningless hybrid.²

As was discussed in regard to "postprocessing," it is more effective to concentrate not on the final outputs of pitch trackers, but on the earlier stages. If different pitch determination algorithms can indeed help each other, then they should do so while they are estimating pitch, not after. After the answer has been computed, all that can be done is to decide whether it is erroneous or not; if the methods inform each other earlier, it might be possible to avoid those errors in the first place. This way it becomes at least conceivable that one would succeed where every pitch tracker would have failed by itself. Also, it now becomes more reasonable to use trackers of different kinds of pitch, since one might arrange it so that only one of them provides the final answer. Of course, by moving into the internals of the algorithms the means of combination will be more idiosyncratic and ad hoc, but as with the case of integrating input information, this is not necessarily a bad thing.

There are several ways that this might be done. Having two or more algorithms working in parallel, constantly helping each out, is unnecessarily complicated. Instead, it would probably be more profitably thought of as a single algorithm which just uses several methods in the process of arriving at its answer. Also, the information flow needn't be bidirectional; one might consider using the output of a crude but fast pitch tracker in a more accurate but usually slower one. There are many instances of "bootstrapping" in pitch detection: knowing the pitch roughly would improve the signal processing (analysis windows, spectrum smoothing bandwidths, etc.), and speed

²For these reasons, if the final pitch track is only going to be displayed, it would probably be preferable to skip this combination and just plot all of the contributing pitch tracks together, because humans are so very good at this kind of integration problem.
up various searches, but of course if the pitch is already known accurately, there would be little point in doing any of that processing in the first place. Usually window sizes and search ranges are chosen according to a fixed global pitch range; if they could instead change as the signal changes the processing would be faster and probably more accurate. Thus one might consider sequentially applying several algorithms—or iteratively apply the same algorithm—and gradually fine tune the pitch estimates, by taking advantage of this bootstrapping capability. Note that with this method, there will probably be little use for the sophisticated control structures previously discussed, because the main purpose of them is to use context to help predict values; if a rough pitch track is available, this is often unnecessary.
Bibliography


