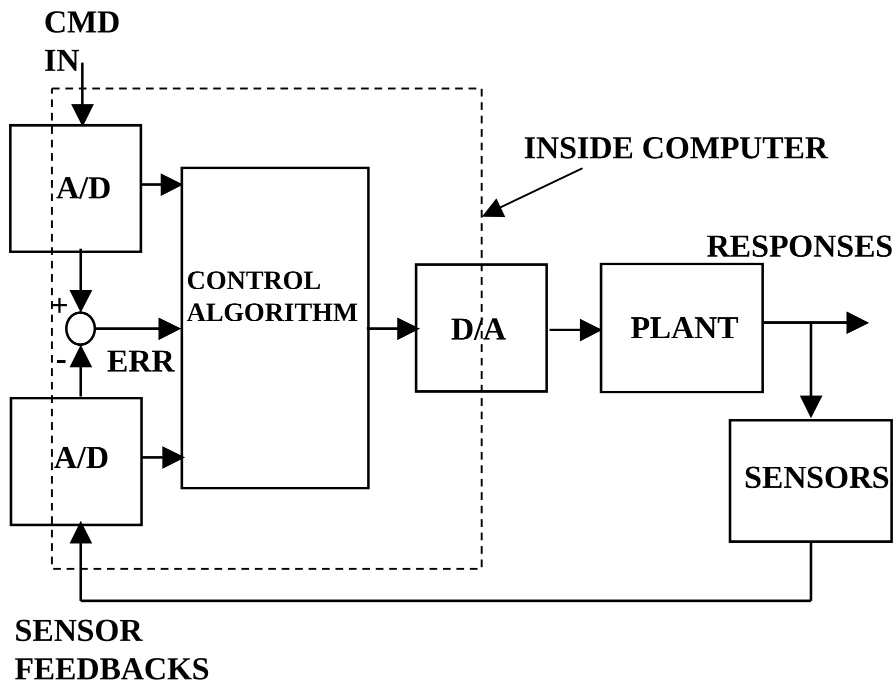

Figure 4.1. Real-time digital control loop. The plant (eg., loudspeaker) is controlled by a control algorithm (eg., vowel synthesizer). User inputs (eg., from a virtual reality glove), sensor feedback (eg., measured noise level), and calculated error signal are input to the control algorithm and must generate updated control outputs within a single sample period

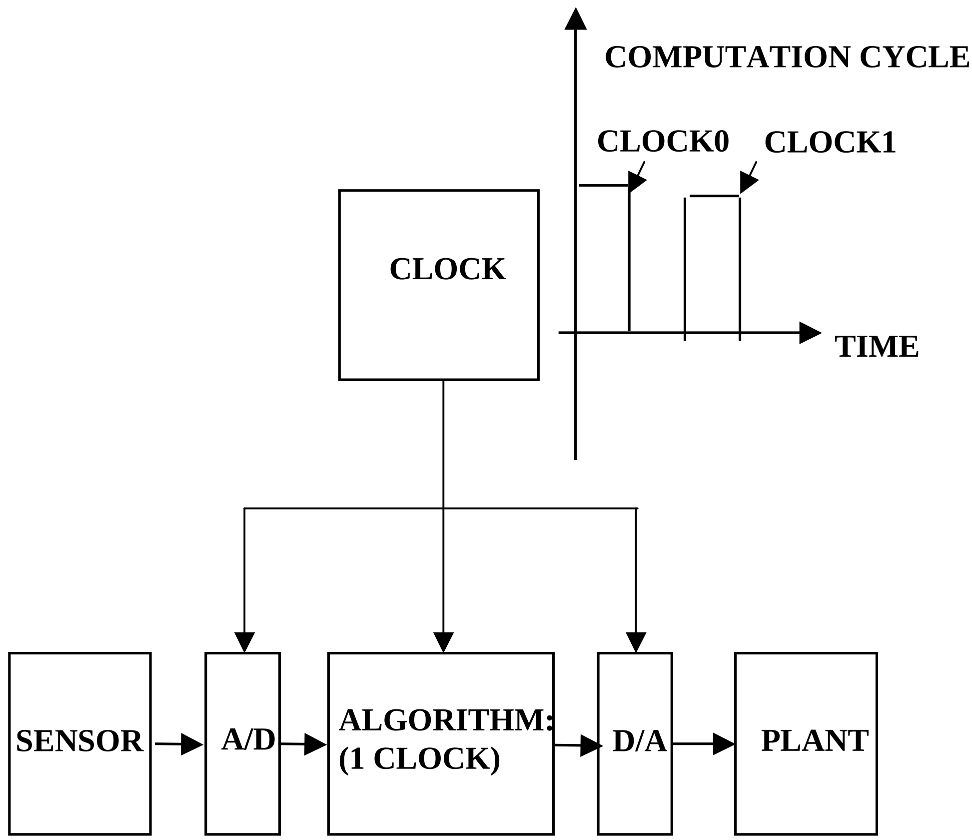

Figure 4.2. Real-time control timing. For real-time digital control, the entire computation cycle must be completed in one sample period. The A/D's provide comand and sensor inputs to the CPU on clock0. The CPU performs control calculations between clock0 and clock1. New D/A values are output to the plant on clock1.

CPU CHIP	CLOCK Mhz	CYCLE/ MULT	uSEC / MULT	uSEC / 10 RESNTR	% 100uS BUDGET
8086	**5.0**	**168**	**33.6**	**1848**	**1848**
80286	**12.5**	**168**	**13.4**	**737**	**737**
80386	**33.0**	**53**	**1.6**	**88**	**88**
80486	**66.0**	**13**	**0.20**	**11**	**11**
80586	**120.0**	**1**	**0.0083**	**0.46**	**0.46**

Figure 4.3. X86 processor performance progression. The numerical processing speed of the PC CPU's increased dramatically over the progression from 8086 to 80586 (Pentium) as both clock speed and processor efficiency improved. Column 2 is CPU clock speed, column 3 is number of clocks to perform a multiplication in the formant resonator calculation process (16 bit fixed point for the 8086 – 80286, and 64 bit floating for the rest). Column 4 is the time in microseconds to perform the multiply. Column 5 is the time to perform all the calculations for ten formant resonators. Column 5 is the percentage of a 10 kHz cycle budget (0.1 ms) consumed by calculations for 10 resonators. It is noteworthy that the transition from 16 bit integer to 64 bit floating actually resulted in *less* CPU time.

ALPHA REAL-TIME SYNTHESIZER FUNCTIONAL OVERVIEW

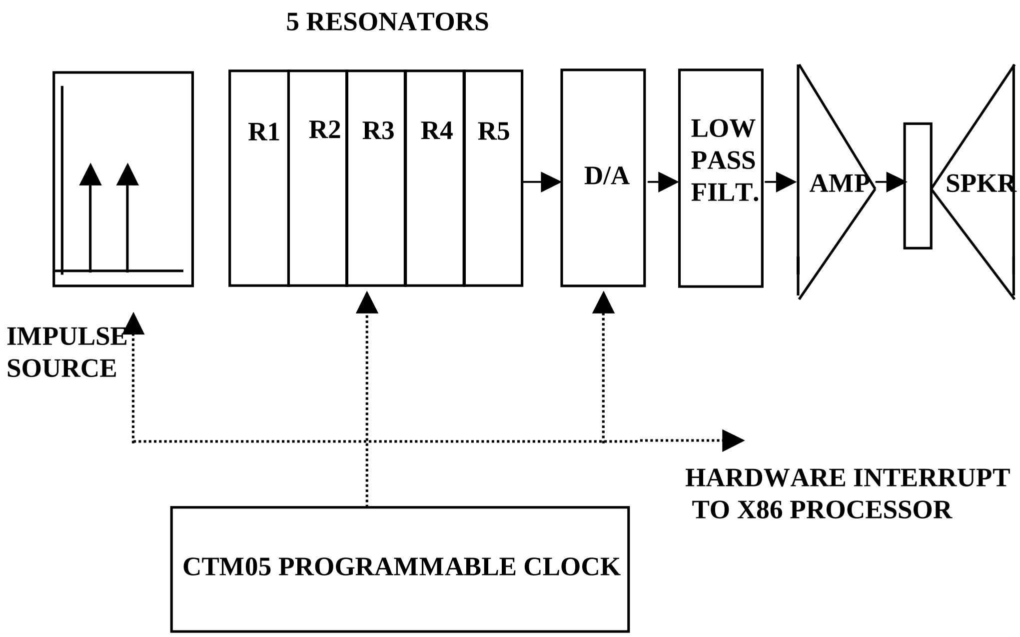

Figure 4.4. Overview of the alpha real-time vowel synthesizer. This first implementation used commercially available adapter cards and prototyping circuitry external to the PC platform. It used a simple impulsive source to excite a bank of 5 second order digital resonators implemented in 16 bit scaled integer arithmetic and programmed in X86 assembly language.

CURRENT REAL-TIME SYNTHESIZER FUNCTIONAL OVERVIEW

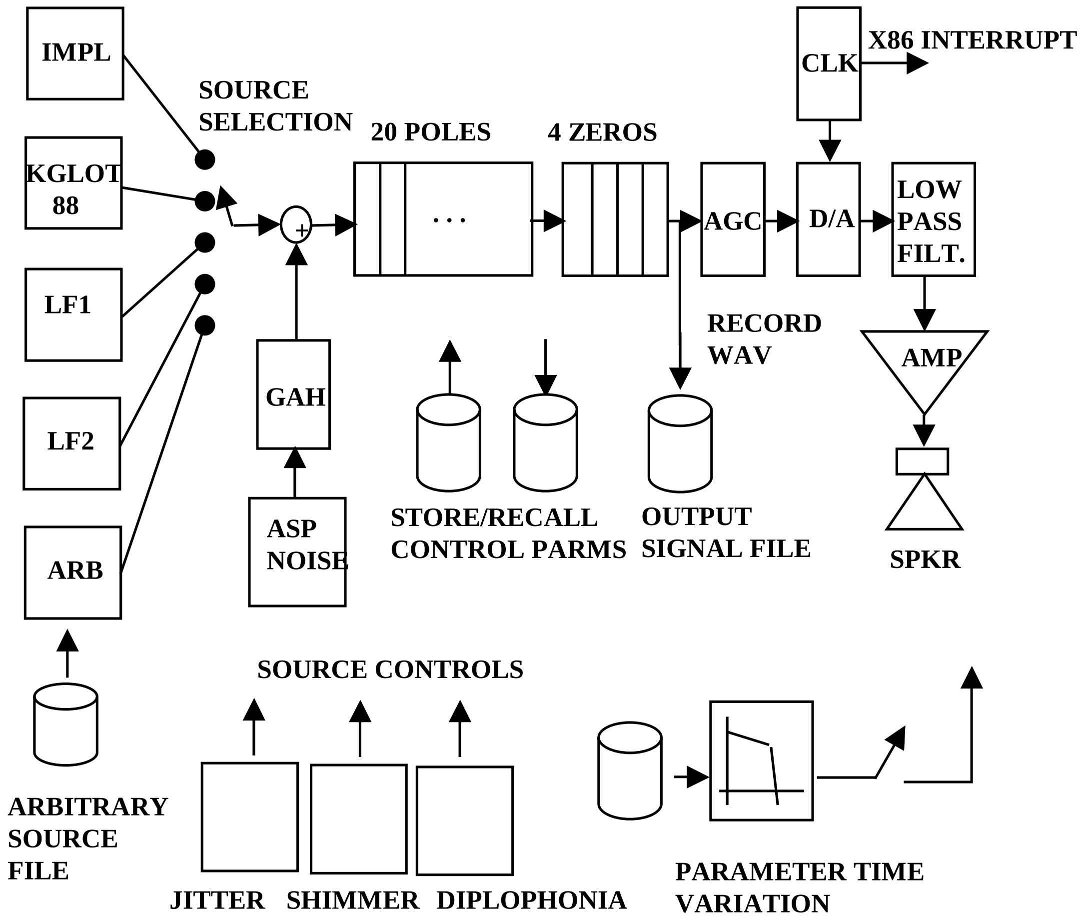

Figure 4.5. Overview of the current real-time synthesizer. Upgrades include flexible source specification (impulse, KGLOTT88, LF, or arbitrary waveform), an aspiration noise source, vocal tract transfer function numerator zeros, an automatic gain control, jitter, shimmer, diplophonia, arbitrary time variation of parameters, and the ability to store and recall time series and parameter sets. All hardware components are grouped on one adapter card.

REAL-TIME SYNTHESIZER HARDWARE

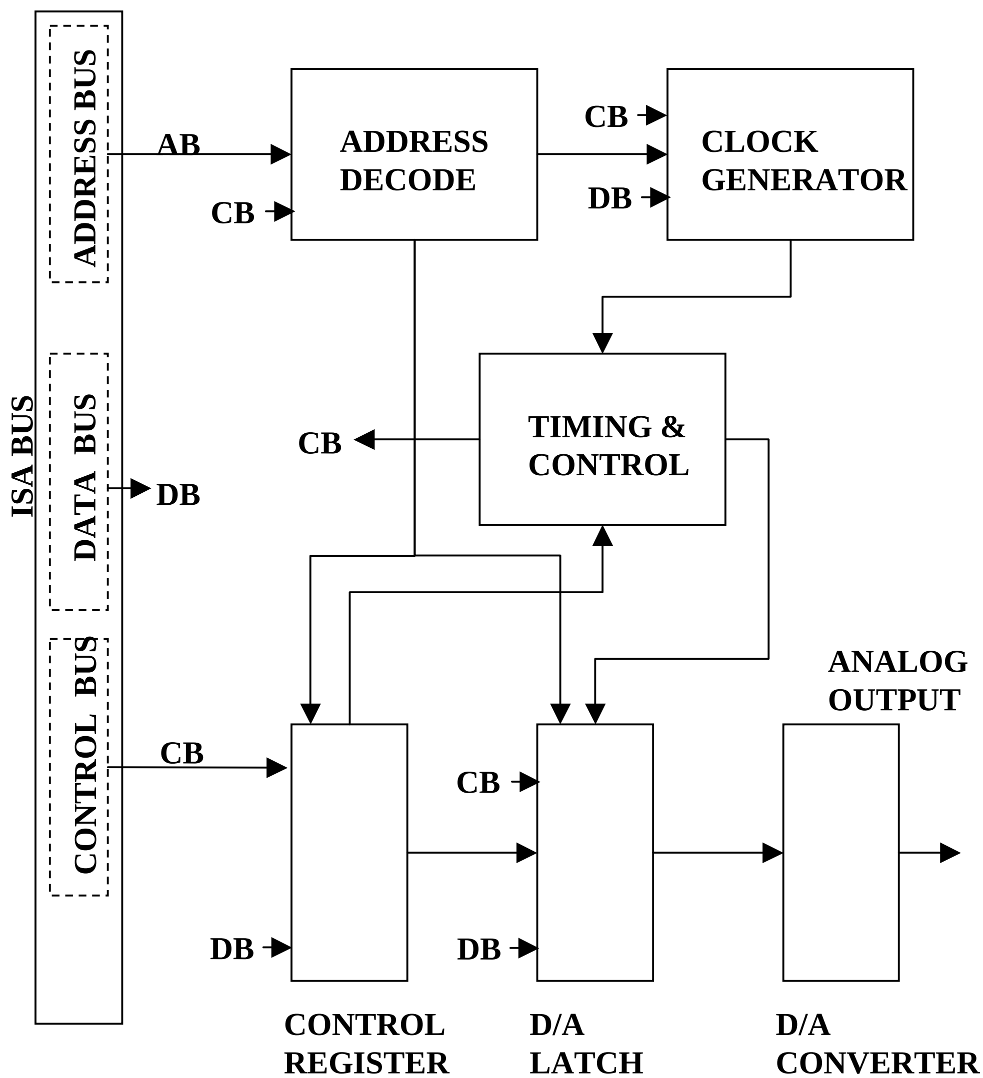

Figure 4.6. Real-time synthesizer hardware. Several hardware features were required to achieve true real-time performance on the PC platform. These include a crystal-controlled clock, a D/A converter, data latches, and timing and control glue logic to interface these components.

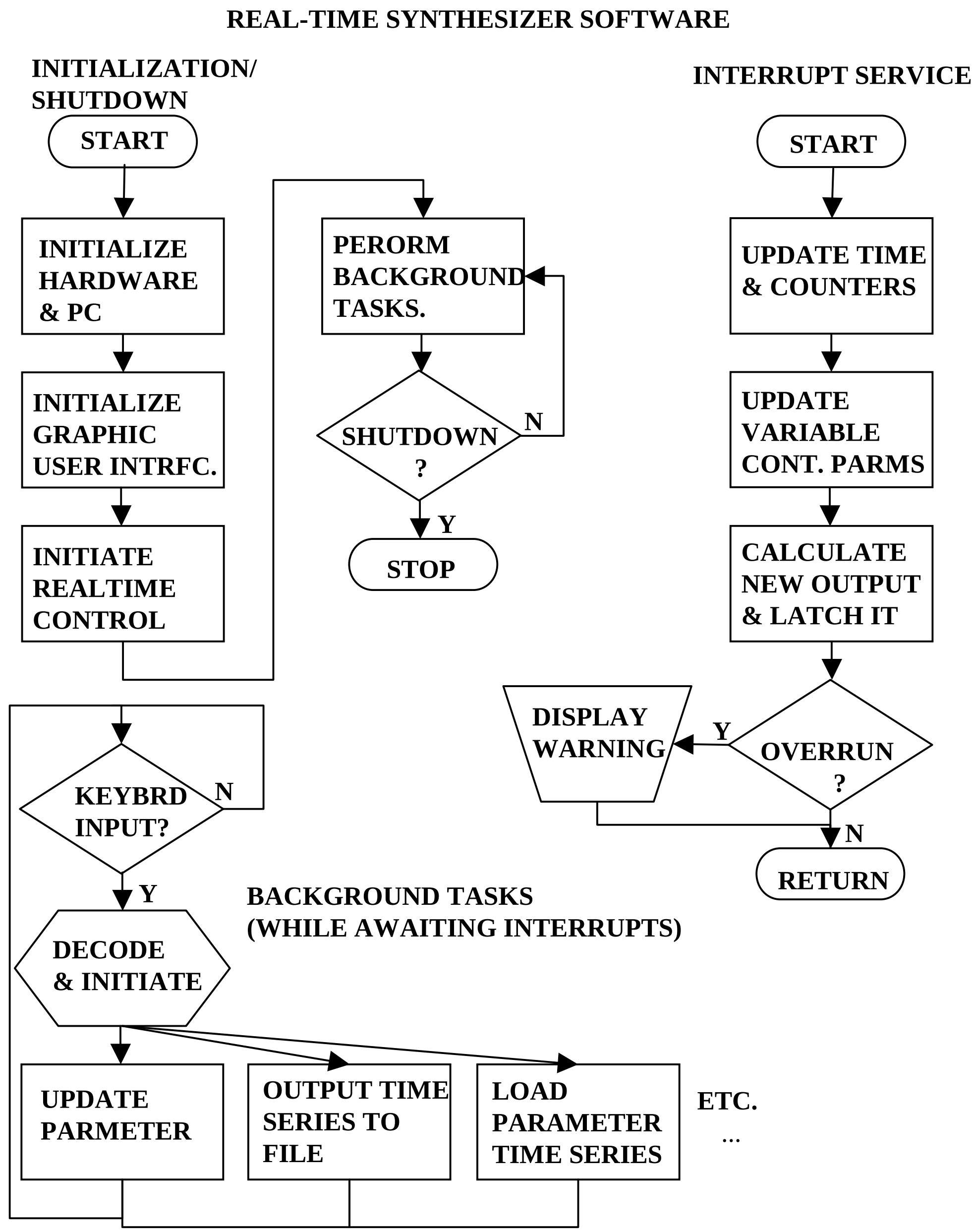

Figure 4.7. Real-time synthesizer software. Programming code may be segregated into two types: foreground (hard real-time tasks), and background (user interface, system management, etc.). The foreground tasks were performed in an ISR (interrupt service routine) and were coded in assembly language for fastest possible execution. The background tasks were coded in C language.

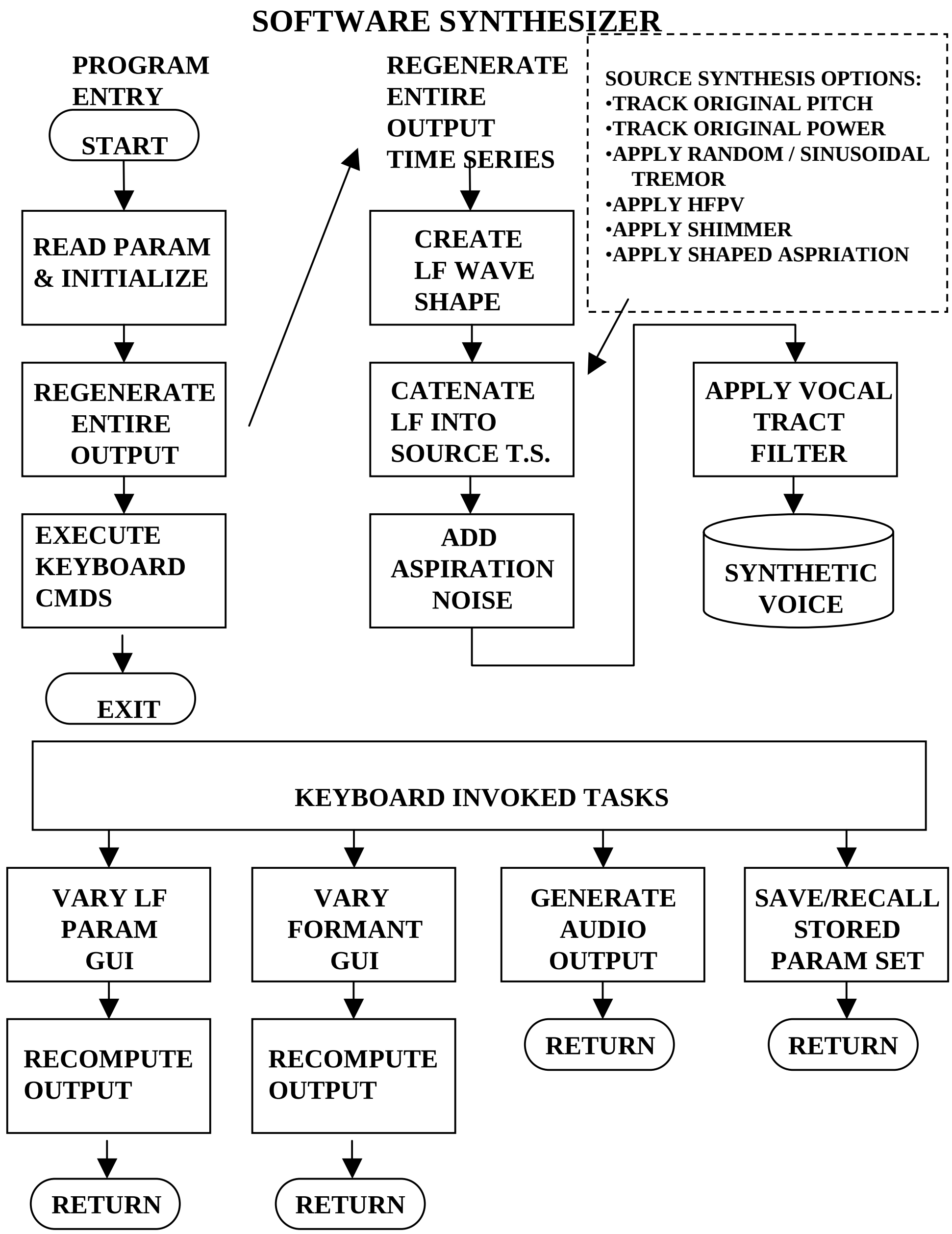

Figure 4.8. Overview of software synthesizer operation. After invocation, the synthesizer loads parameters for the requested (previously analyzed) case and calculates the synthetic version. The program then waits for user input commands to execute functions such as modifying the LF source parameters, modifying formants, or dumping or loading time series to/from disk.

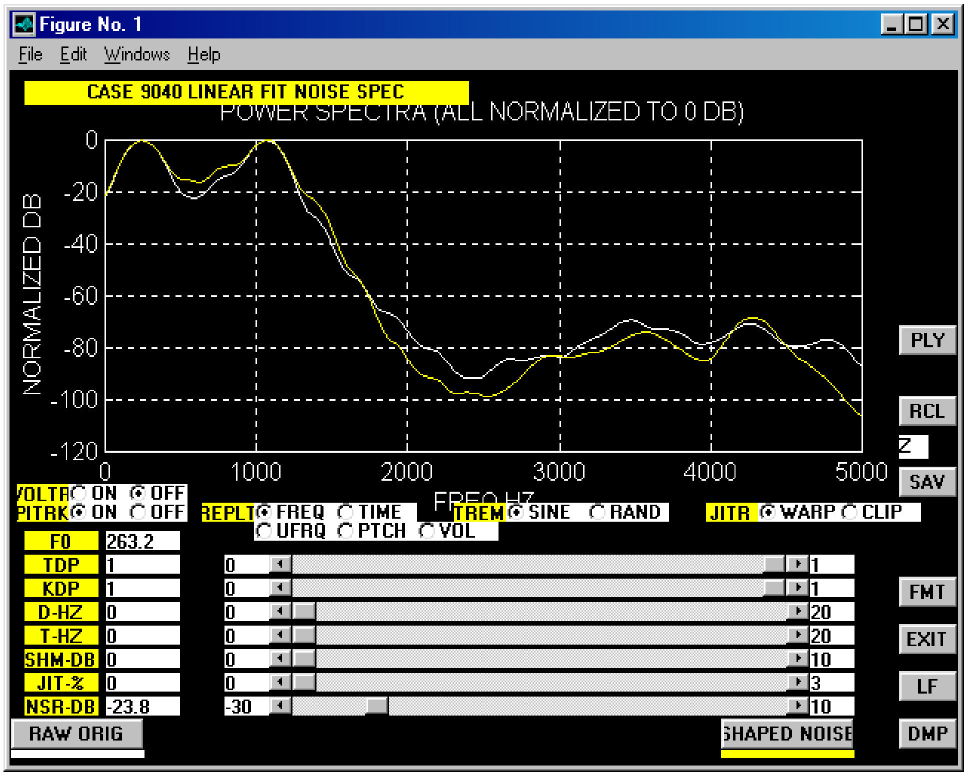

Figure 4.9a. Main GUI (graphical user interface) for the software synthesizer. Provisions are included for user specification via sliders or text of noise levels, HFPV, shimmer, etc. Options are included for tasks such as activating playback of the original or synthetic voice, turning on/off fundamental frequency/volume tracking, plotting spectra or time series, or invoking modification of LF or formants.

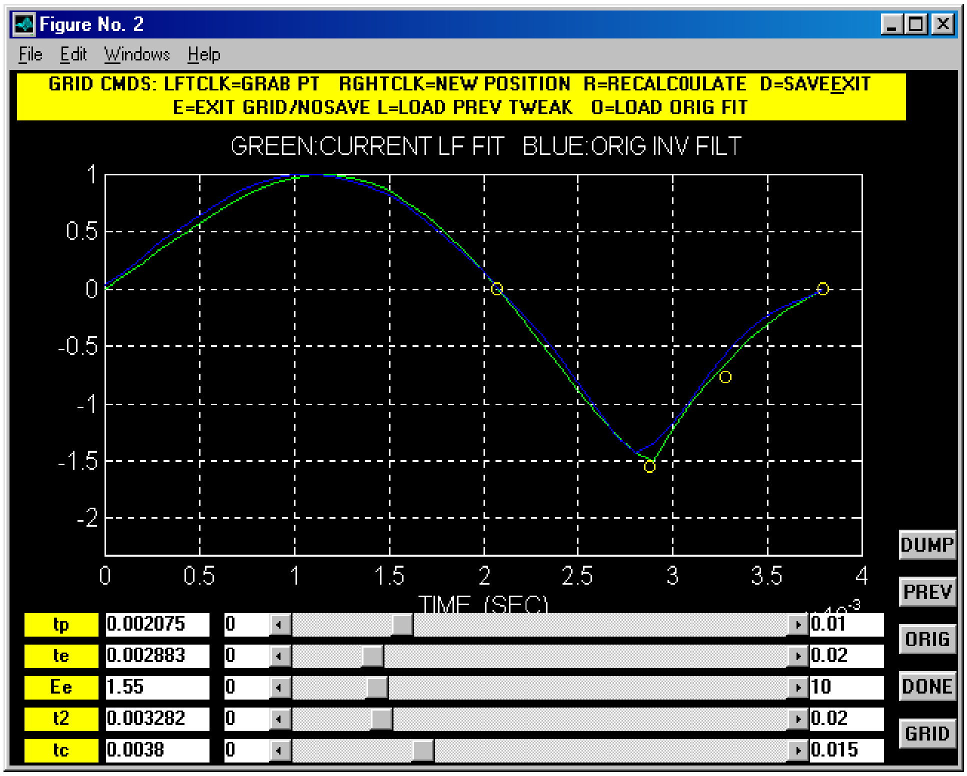

Figure 4.9b. LF modification GUI. This screen allows the user to control the shape of the LF source waveform by varying the LF parameters.

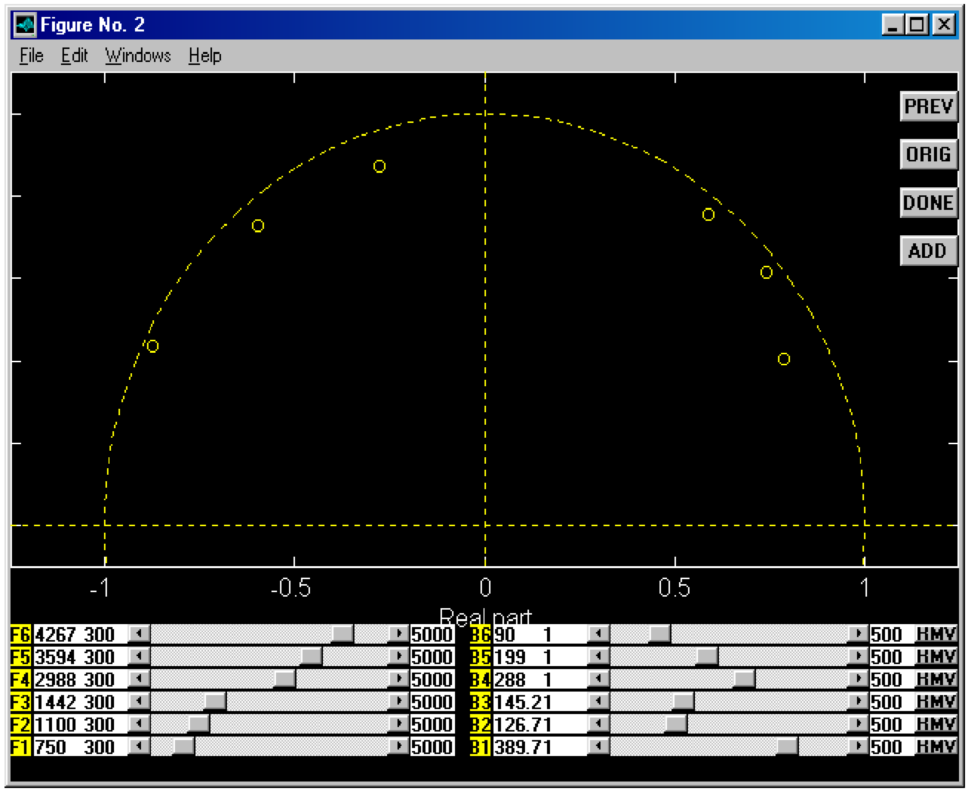

Figure 4.9c. Formant modification GUI. This screen allows the user to move the pole locations specifying the all-pole vocal tract model.