Homer Autotuner: Selected Topics

Table of Contents

1. Introduction

This application note discusses the following topics regarding S-TEAM Homer Autotuners:

- Matching range
- Tuning speed

These quantities can be examined using the **HoSim** simulation app after customizing the simulated Autotuner by means of loading the **Tun.mem** tuner characterization file pertaining to a particular Autotuner model.

The **HoSim** simulation software is a part of your Autotuner installation. The latest version can be downloaded from https://s-team.sk/software, please follow the link **HoSim**.

Tun.mem files for a selection of typical S-TEAM Autotuner variants are included in your **HoSim** installation. Their names reflect the Autotuner waveguide type and, where appropriate, the motors speed category, such as **TunR26_Fast.mem**.

A **Tun.mem** file for your particular Autotuner unit can be found under the same name in your installation folder.

2. Customizing Simulated Autotuner

To customize a simulated Homer Autotuner, please follow the steps below:

1. Start **HoSim** app.
2. Open the **System Info** window (Fig. 1) by clicking **View|System Info** menu item, or **Sys** toolbar button.
3. Click **Load Mem** button in **System Info** window. Navigate to the **Tun.mem** file for the tuner of interest (as noted above, it may have a different name, such as **TunR26_Standard.mem**) and open it.

Now the tuner has been customized. Please note the updated settings in *Tuner*, *Motors*, and *Sketch* pages of the **System Info** window.

Now, after terminating and restarting **HoSim**, the tuner parameters will assume their latest settings, and hence there is no need to reload the **Tun.mem** again.

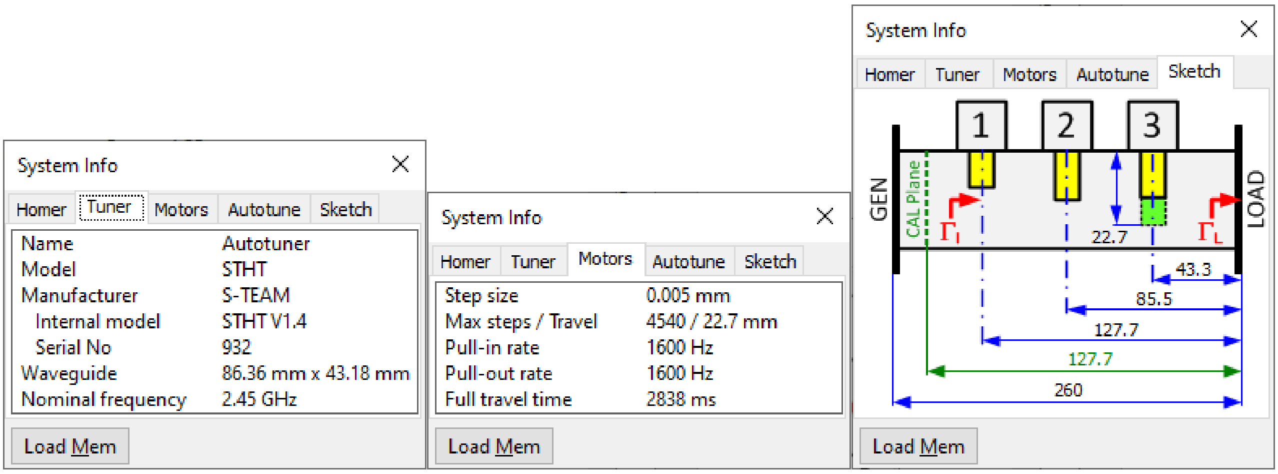

Fig. 1. *Tuner*, *Motors*, and *Sketch* pages of the **System Info** window of **HoSim** app.

3. Setting Measurement Conditions

To set measurement conditions, proceed as follows. Note that the settings will not be active until clicking **Apply** button.

- Open the **Homer Model** window (Fig. 2) by clicking the menu item **Model**.

Fig. 2. Pages of **Homer Model** window.

- Click the **DUT** tab to define type and value of the load reflection coefficient $\Gamma_L = M \exp(j\varphi)$. Please note that by default Γ_L refers to the Autotuner output flange (see the *Sketch* page of Fig. 1).
  - In **Type** box, select the method of simulating the load reflection coefficient Γ_L:
    - *Defined*: Γ_L is defined by its **Magnitude** (M) and **Phase** (φ) edit boxes.
    - *Match*: Load reflection coefficient will be set zero: $|\Gamma_L| = M = 0$.
    - *Animated*: Time-varying reflection coefficient, spiraling in polar diagram according to settings in **Animation** page. For details, refer to Help for Homer, topic *Homer Simulation > Homer Model Window > Animation Page* (try the settings shown in Fig. 2).

- *Chart*: Using a mouse click or dragging to set a load reflection coefficient in polar diagram (a resizable polar diagram opens after selecting **Chart** and clicking **Apply** button). For details, refer to Help for Homer, topic *Homer Simulation* > *Homer Model Window* > *Chart*.

– Define the reflection coefficient uncertainty radius (which simulates noise) by editing the **Uncertainty Radius** edit box. Set zero for an ideal noiseless situation. Note that increasing averaging in CW sampling mode (e.g., via Sample Count button) reduces the observed noise. Thus, to see the full uncertainty, set Averaging = 1.

- Click **Freq/Power/Temp** tab to define the following simulated quantities:
  – Generator frequency and its uncertainty in %;
  – Generator incident power and its uncertainty (noise).
  – Autotuner internal temperature and its uncertainty in degrees Celsius.
  – Two of the error flags delivered by the system (Overflow, Low Signal).
- Click **Apply** to activate the settings.

The **Homer Model** window can be kept open while running the measurement, thus enabling any changes to be applied "live."

4. Matchable Area

The area of matchable load reflection coefficients Γ_L for a given Autotuner at a given frequency can be viewed in the polar diagram by following these steps:

1. Customize the Autotuner.
2. Set a desired magnetron frequency in the **Homer Model** window (Fig. 2).
3. Set the DUT type to *Chart* and click Apply button in the **Homer Model** window. The **Load Reflection Coefficient** chart opens (Fig. 3).
4. In **Show Area** box, select *Matchable*. Two subareas of the reflection coefficients matchable by the stubs 1 + 2 (the blue-bordered area) and by the stubs 2 + 3 (the pink-bordered area) will be drawn. Note that the reflection coefficients are related to the reference plane as discussed above.

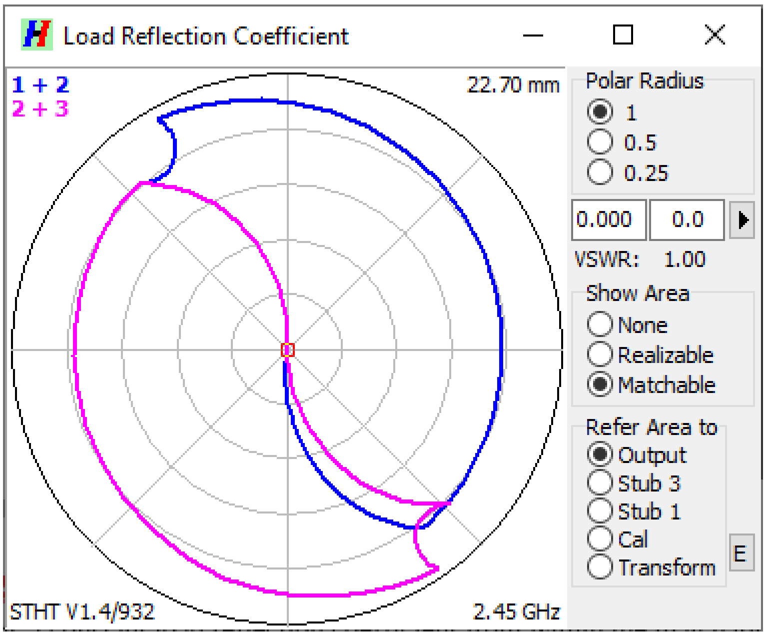

Fig. 3. **Load Reflection Coefficient** window displaying the matchable area.

Now, you can change frequency in the **Homer Model** window (activate it by clicking **Apply**) and observe how the matchable area will change.

You can also experiment with the autotuning by using the following procedure:

- Open the **Tuner View** window (Fig. 4) by clicking the menu item **Tuner|View**, or the toolbar button .

- Activate autotuning by depressing **AT** button in **Tuner View** window.
- Start measurement.
- Click into the **Load Reflection Coefficient** window, move the cursor in its polar chart and observe the measured reflection coefficient and the insertions of the tuning stubs.

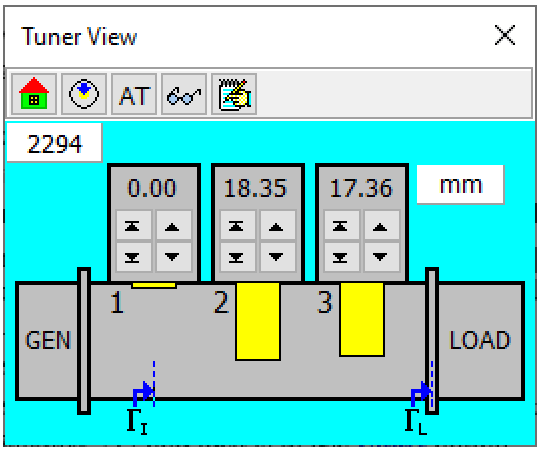

Fig. 4. **Tuner View** window of the **HoSim** program.

5. Tuning Speed

Let us suppose for the moment that

– the load reflection coefficient Γ_L is constant,
– Γ_L lies inside the tuner matchable area,
– generator (magnetron) frequency is constant.

The process during which the tuner input reflection coefficient reaches a target value (normally zero) within a given tolerance will be referred to as *autotuning sequence*. One autotuning sequence consists of one or more *autotuning steps*. One autotuning step consists of the following actions, with their respective times:

1. Measurement time t_m
2. Data processing time t_p
3. Tuning stubs movement time t_s
4. Writing time of final motor insertions to flash memory t_w

Since S-TEAM autotuners employ predictive algorithms, *one* autotuning step is ideally sufficient to complete each autotuning sequence. However, in practice, due to nonidealities, more than one autotuning step is sometimes necessary (a *basic* step plus one or two corrections with progressively shorter stub movements). These “nonidealities” include systematic and random measurement errors, and most importantly, magnetron signal impurity and instability. The latter, fairly unpredictable, factor is often dictated by a particular anode high-voltage generation scheme. Due to these uncertainties, which vary from system to system, it is meaningful to measure the tuning speed performance merely by the *basic* autotuning step, and assuming a good quality (low-ripple, harmonic) signal. Some extra time (overhead) to account for corrective steps can then be added. Under these assumptions, the tuning speed performance can be expressed in terms of a single-step autotuning time t_t as

$$t_t = t_m + t_p + t_s + t_w \qquad (1)$$

The four tuning time constituents (t_m, t_p, t_s, t_w) play unequally significant roles.

5.1 Measurement Time

Measurement process can be user-configured such that the measurement time t_m varies widely, ranging typically from 100 μs to several seconds. Because of this variability, it is reasonable to ***not*** consider the measurement time for the assessment and comparison of Autotuner speed performance. For practical assessments, users can correct the tuning time by adding their individually estimated measurement times t_m.

For **CW** sampling mode, measurement time can be estimated as

(2) $$t_m = \frac{N_s + 15}{f_s} + 4.5 + 0.0448\left(N_s + 16\right)$$

where

- N_s is sample count (averaging number),
- f_s is sampling frequency.

For **Rectified** sampling mode, measurement time can be estimated as

(3) $$t_m = 16 + 2\frac{N_R}{f_R} + 0.169 N_s$$

where

- f_R is ripple frequency,
- N_R is number of sampled ripple periods,
- N_s is sample count.

For **Pulsed** sampling mode, the measurement time can be estimated as

(4) $$t_m = 254 + T_{\max} + T_s + 0.111\, N_s$$

where

- T_s is sampling time
- N_s is sample count
- T_{max} is maximal expected pulse repetition period.

5.1.1 Triggering

In case of **Rectified** and **Pulsed** modulations, and hence the corresponding appropriate sampling modes, waiting time for the occurrence of triggering events should also be considered. This time can be anywhere between zero and $2 \times T_{trg}$ where $T_{trg} = 1/f_R$ for the Rectified mode and $T_{trg} = 1/f_p$ for the Pulsed mode, with f_p being the pulse modulation repetition rate. The factor 2 appears because in one measurement cycle, signal sampling and frequency counting are triggered successively.

5.2 Data Processing Time

Data processing time t_p includes computation of new tuning stubs positions. It takes typically less than 10 ms, which constitutes only a fraction of the total tuning time. It can be therefore ignored ($t_p \approx 0$).

5.3 Tuning Stubs Travel Time

The value of the tuning stubs travel time t_s depends on the stepper motors type and the distances the stubs must travel. Since the stubs move simultaneously, the longest of the three travel distances is relevant.

The basic stepper motor/spindle combination characteristics are

- Step size s (mm)
- Stepping rate f_s (Hz)

The values can be learned from the **Motors** page of the **System Info** window shown in Fig. 1 (take the lower of the *Pull-in rate* and *Pull-out rate* values for f_s). The window also shows the maximum *Travel* distance (insertion depth) $h_{\max}$, which is linked with *Max steps* ($n_{\max}$) value by

(5) $$h_{\max} = s n_{\max}$$

The stub travel speed is

(6) $$v_s = s f_s$$

Travelling a distance Δh takes the time[1]

$$t_s = \frac{\Delta h}{v_s} = \frac{\Delta h}{s f_s} \qquad (7)$$

5.4 Flash Writing Time

Some applications need the stubs to remain in positions even after switching the Autotuner OFF and later ON again. For this reason, the stub positions (in terms of motor steps) are written into the Autotuner internal flash memory after each stubs movement command. The process takes approximately 200 ms, hence

$$t_w = 0.2 \qquad (8)$$

The flash writing time is therefore a significant limiting factor in fast Autotuners.

5.5 Worst Case Tuning Time Estimate

The tuning time estimates below are based on the following assumptions:

1. Starting stub positions are zero (stubs completely withdrawn). The travel distance Δh then equals the stub insertion h.
2. Based on the arguments above, the measurement time t_m is not included, and data processing time t_p is neglected.

The worst case occurs for load reflection coefficients on the border of the matchable area (Fig. 3) or beyond because at least one of the stubs must move to the maximum insertion depth $h_{\max}$, resulting in the longest possible travel time

$$t_{s\max} = \frac{h_{\max}}{v_s} \qquad (9)$$

Such extreme mismatch cases are also more sensitive to nonidealities, and hence, in practice, one or two additional corrective steps are typically required. As suggested above, this can be accounted for by adding some overhead, which may be conveniently expressed as percentage u of the basic step value (9). The formula for the worst-case tuning time is then

$$t_{t\max} = \frac{h_{\max}}{v_s}\left(1 + \frac{u}{100}\right) + t_w \qquad (10)$$

where u is the overhead factor in percent (e.g., u = 50%). Examples for typical tuner models are shown in Table 1.

5.6 Partial Mismatch

The above formula can also be used for estimating the tuning time in the case of partial mismatch (load reflection coefficient inside the matchable area). In this case, $h_{\max}$ should be replaced by the corresponding stub insertion h, and hence

$$t_t = \frac{h}{v_s}\left(1 + \frac{u}{100}\right) + t_w \qquad (11)$$

The stub insertion h needed for matching depends on both the magnitude M and phase φ of the load reflection coefficient $\Gamma_L = M \exp(j\varphi)$. The dependence on phase is less significant and can be ignored for such estimates. A fairly useful approximation is then

$$h = \frac{h_{\max}}{2}\left[1 + \frac{M}{M_{\max}} - \exp\left(-20\frac{M}{M_{\max}}\right)\right] \qquad (12)$$

---

[1] Motor acceleration and deceleration have been neglected.

$$t_t = \frac{h_{max}}{2v_s}\left[1+\frac{M}{M_{max}}-\exp\left(-20\frac{M}{M_{max}}\right)\right]\left(1+\frac{u}{100}\right)+t_w \qquad (13)$$

where M_{max} is the maximum load reflection coefficient magnitude matchable irrespective of phase, i.e., the radius of the largest circle that completely fits into the matchable area (in Fig. 3 about 0.78). For $M > M_{max}$, use $h = h_{max}$.

The curve (12) is presented in Fig. 5. As evident, it is very steep close to the origin: matching of even small reflections requires substantial stub insertions, and hence comparatively high tuning times. For higher *M*, the curve becomes less steep.

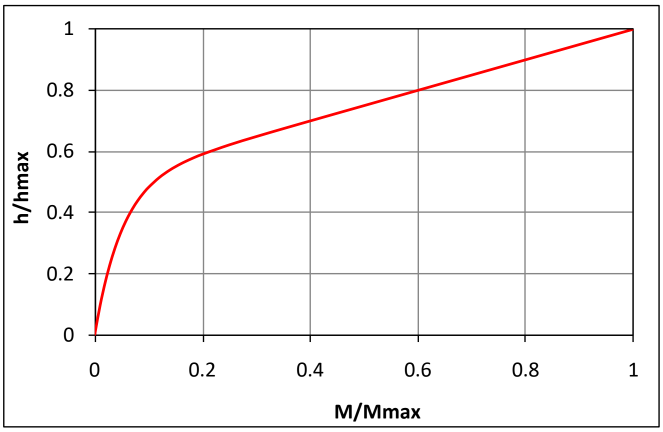

Fig. 5. Relative stub insertion depth *h* vs. relative load reflection coefficient magnitude *M*.

Computations for a variety of Autotuner models can be performed using the accompanying Excel worksheet **AN0901.xlsx**. An example is shown in Table 1.

Table 1. Tuning time computations for a selection of autotuners.

Autotuner Designation			**Standard R26**	**Fast R26**	**R9 Stopa**	**R58**
Model			**STHT 1.8 / 1.9**	**STHT 1.8 / 1.9**	**STHT 2.3**	**STHT 3.1**
Max travel	h_max	mm	22.7	22.7	69.977	11
Max matchable magnitude	M_max	mm	0.78	0.78	0.84	0.82
Motor/Spindle						
Step size	s	mm	0.005	0.1	0.127	0.02
Stepping rate	f_s	Hz	1600	1000	1000	800
Derived Quantities						
Steps/mm		1/mm	200	10	7.87	50
Stub speed	v_s	mm/s	8	100	127	16
Max steps	n_max		4540	227	551	550
Full travel time	ts_max	s	2.84	0.23	0.55	0.69
Tuning Time Estimate			*Inputs*			
Tuning time overhead	u	%	*50*	*50*	*50*	*50*
Flash write time	tw	s	0.2	0.2	0.2	0.2
Worst case tuning time	tt_max	s	**4.46**	**0.54**	**1.03**	**1.23**
Load reflection coefficient magnitude	M		*0.1*	*0.1*	*0.1*	*0.1*
Travel distance	h	mm	11.93	11.93	35.92	5.69
Steps	n		2386	119	283	285
Travel time	ts	s	**1.49**	**0.12**	**0.28**	**0.36**
Tuning time	tt	s	**2.44**	**0.38**	**0.62**	**0.73**

5.7 Using HoSim Simulation App

The **HoSim** Homer Autotuner simulation app is a best tool for more accurate estimation of the tuning time. It can model a tuning

- for any reflection coefficient magnitude and phase,
- at any desired frequency,
- with any starting stub positions.

To do so, proceed as follows:

1. Customize the Autotuner.
2. Open the **Homer Model** window (Fig. 2) by clicking the menu item **Model**.
3. Set the desired magnetron frequency in the **Homer Model** window, page *Freq/Power/Temp* (Fig. 2).
4. Set the desired DUT reflection coefficient magnitude and phase by any method described in Setting Measurement Conditions (except *Animated*). A good idea is to use the *Chart* option.
5. Open the **Tuner View** window (Fig. 4) by clicking the menu item **Tuner|View** or the toolbar button. Observe that unlike the "sharp" **HomSoft** program, the window includes the **Travel Time** display box (top left, showing the number 2294).
6. Disable continuous autotuning by setting **AT** to the non-depressed position.
7. Set the stubs to desired starting insertions (e.g., by dragging, using the up-down buttons, or clicking the insertion numbers).
8. Start measurement by clicking in the main **HoSim** window.
9. Activate a single autotuning step by clicking **Tuner View** window the **Travel Time** box or button.
10. Notice the number in the **Travel Time** box. The figure represents the stubs travel time t_s in milliseconds.
11. You may wish to add some overhead and the flash writing time t_w (200 ms) as discussed in the previous section to estimate the overall tuning time.
12. Repeat the whole procedure for any desired conditions. Try adding in **Homer Model** window an uncertainty of reflection coefficient and/or frequency.
13. To estimate the overall tuning time more realistically, you may wish to add other constituents as discussed in the previous sections: a measurement time t_m, the flash writing time t_w (200 ms), potential triggering wait times $2 \times T_{trg}$ as well as some overhead.