Homer Autotuner: Selected Topics

Table of Contents

[1. Introduction 1](#_Toc168652659)

[2. Customizing Simulated Autotuner 1](#_Toc168652660)

[3. Setting Measurement Conditions 2](#_Toc168652661)

[4. Matchable Area 3](#_Toc168652662)

[5. Tuning Speed 4](#_Toc168652663)

[5.1 Measurement Time 4](#_Toc168652664)

[5.1.1 Triggering 5](#_Toc168652665)

[5.2 Data Processing Time 5](#_Toc168652666)

[5.3 Tuning Stubs Travel Time 5](#_Toc168652667)

[5.4 Flash Writing Time 6](#_Toc168652668)

[5.5 Worst Case Tuning Time Estimate 6](#_Toc168652669)

[5.6 Partial Mismatch 6](#_Toc168652670)

[5.7 Using HoSim Simulation App 8](#_Toc168652671)

[6. Feeding an Applicator via Multiple Ports 8](#_Toc168652672)

# 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 application after customizing the simulated Autotuner by 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>.

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.

# 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 therefore there is no need to reload the Tun.mem.

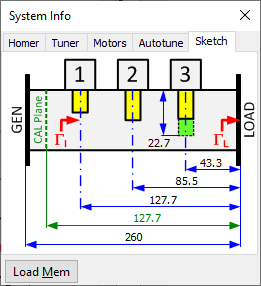
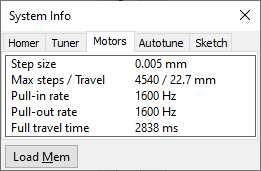
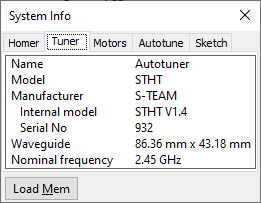
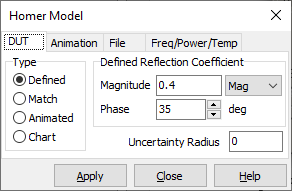
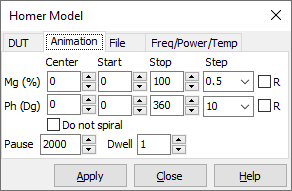


Fig. 1. *Tuner*, *Motors,* and *Sketch* pages of the **System Info** window of **HoSim** app.

# Setting Measurement Conditions

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

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

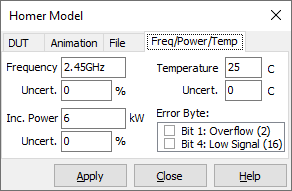


Fig. 2. Pages of the **Homer Model** window.

* Click the **DUT** tab to define the type and value of the load reflection coefficient Γ*L* = *M* exp(*jϕ*). Please note that by default Γ*L* refers to the Autotuner output flange (see the *Sketch* page of Fig. 1).
* In the **Type** box, select the method of simulation for the load reflection coefficient Γ*L*:
* *Defined*: Γ*L* is defined by its **Magnitude** (*M*) and **Phase** (*ϕ*) edit boxes.
* *Match*: The load reflection coefficient is set to zero: |Γ*L*| = *M* = 0.
* *Animated*: A time-varying reflection coefficient, spiraling in the polar diagram according to settings in the **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*: The load reflection coefficient is set using a mouse click or dragging in the polar diagram (a resizable polar diagram opens after selecting **Chart** and clicking the **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 to zero for an ideal noiseless situation. Note that increasing averaging in CW sampling mode (e.g., via Sample Count button sbNsampl ) reduces the observed noise. Thus, to see the full uncertainty, set Averaging = 1.
* Click the **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.”

# 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](#_Autoexec) the Autotuner.
2. Set a desired magnetron frequency in the **Homer Model** window (Fig. 2).
3. Set the DUT type to *Chart* and click the **Apply** button in the **Homer Model** window. The **Load Reflection Coefficient** chart opens (Fig. 3).
4. In the **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](#_Setting_Measurement_Conditions).

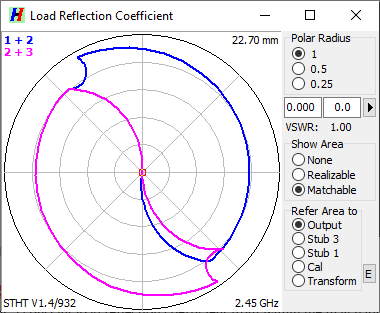


Fig. 3. The **Load Reflection Coefficient** window displaying the matchable area.

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 Tuner.
* Activate autotuning by depressing the **AT** button in the **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.

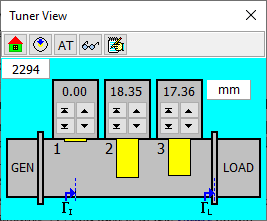


Fig. 4. The **Tuner View** window of the **HoSim** program.

# 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,
* the generator (magnetron) frequency is constant.

The process during which the tuner input reflection coefficient reaches a target value (usually zero) within a given tolerance will be referred to as an *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 *tm*
2. Data processing time *tp*
3. Tuning stubs movement time *ts*
4. Writing time of final motor insertions to flash memory *tw*

Since S-TEAM autotuners employ predictive algorithms, *one* autotuning step is ideally sufficient to complete each autotuning sequence. However, in practice, due to nonidealities, multiple autotuning steps are often 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 only by the *basic* autotuning step, and to assume 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 *tt* as

( 1 ) 

The four tuning time constituents (*tm*, *tp*, *ts*, *tw*) play unequally significant roles.

## Measurement Time

The measurement process can be configured by the user such that the measurement time *tm* varies widely, typically ranging 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 *tm*.

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

( 2 ) 

where

* *Ns* is sample count (averaging number),
* *fs* is sampling frequency.

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

( 3 ) 

where

* *fR* is ripple frequency,
* *NR* is number of sampled ripple periods,
* *Ns* is sample count.

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

( 4 ) 

where

* *Ts* is sampling time
* *Ns* is sample count
* *Tmax* is maximal expected pulse repetition period.

### Triggering

In the case of **Rectified** and **Pulsed** modulations, as well as their 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 × *Ttrg* where *Ttrg* = 1/*fR* for the Rectified mode and *Ttrg* = 1/*fp* for the Pulsed mode, with *fp* being the pulse modulation repetition rate. The factor 2 appears because in one measurement cycle, signal sampling and frequency counting are triggered successively.

## Data Processing Time

The data processing time *tp* includes computation of new tuning stubs positions. It typically takes less than 10 ms, which constitutes only a fraction of the total tuning time. It can therefore be ignored (*tp* ≈ 0).

## Tuning Stubs Travel Time

The value of the tuning stubs travel time *ts* 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 *fs* (Hz)

These 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 *fs*). The window also shows the maximum *Travel* distance (insertion depth) *h*max, which is linked with the *Max steps* (*n*max) value by

( 5 ) 

The stub travel speed is

( 6 ) 

Travelling a distance Δ*h* takes the time[[1]](#footnote-1)

( 7 ) 

## 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, and thus

( 8 ) *tw* = 0.2

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

## 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 Δ*h* then equals the stub insertion *h*.
2. Based on the arguments above, the measurement time *tm* is not included, and data processing time *tp* is neglected.

The worst case occurs for load reflection coefficients either on the border of or beyond 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

( 9 ) 

Such extreme mismatch cases are also more sensitive to nonidealities, and thus, 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 a percentage *u* of the basic step value ( 9 ). The formula for the worst-case tuning time is then

( 10 ) 

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

## 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 therefore

( 11 ) 

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

( 12 ) 

( 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.

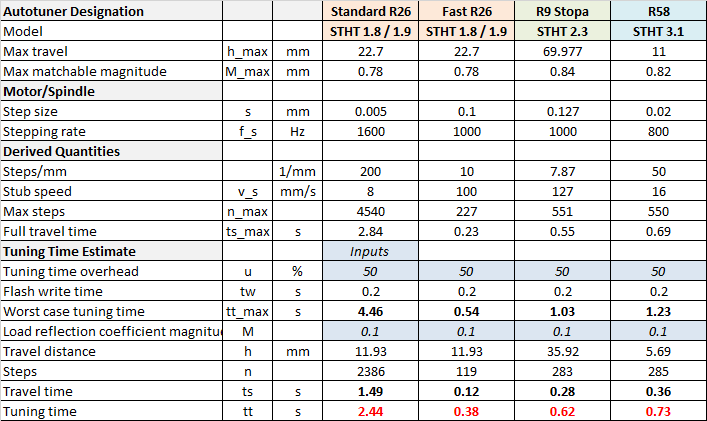
The curve ( 12 ) is presented in Fig. 5. As evident, it is very steep close to the origin: matching of even small reflections requires substantial stub insertions, and hence comparatively high tuning times. For higher *M*, the curve becomes less steep.



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.



## Using HoSim Simulation Application

The **HoSim** Homer Autotuner simulation application is the 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](#_Autoexec) 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](#_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 Tuner toolbar button. Observe that unlike the “sharp” **HomSoft** program, the window includes the **Travel Time** displaybox (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 sbPlay0 in the main **HoSim** window.
9. Activate a single autotuning step by clicking the **Tuner View** window the **Travel Time** box or the sbTune1 button.
10. Notice the number in the **Travel Time** box. The figure represents the stubs travel time *ts* in milliseconds.
11. You may wish to add some overhead as well as the flash writing time *tw* (200 ms) as discussed in the previous section to estimate the overall tuning time.
12. Repeat the whole procedure for any desired conditions. In the **Homer Model** window, try adding 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](#_Measurement_Time) *tm*, the flash writing time *tw* (200 ms), potential [triggering wait times](#_Triggering) 2 × *Ttrg* as well as some overhead.

# Feeding an Applicator via Multiple Ports

If an applicator is fed simultaneously from more than a single port and each of the feeding ports includes an autotuner, then excessive crosstalk between the ports can affect reflection coefficients perceived by the autotuners, and, as a consequence, impair tuning, or even to lead to its instability (constant, seemingly random movement of the tuning stubs).

To estimate a mimimum acceptable isolation between the ports for a satisfactory tuning, consider an applicator in the form of a multiport with *N* > 1 ports, and with scattering parameters *Sij* (*i*, *j* = 1, 2,… *N*). The cross-coupling coefficients are those with *i* ≠ *j*.

Let the waves entering the ports be *ai* , *i* = 1, 2,… *N*. Their magnitudes are related with the powers *Pi* of the sources feeding the ports via

( 14 ) 

Let us analyze the wave *b*1 emerging from port 1 (the argument is similar for all of the ports). Neglecting all but the first-order contributions[[2]](#footnote-2), which we can safely do in targeted applications, the emerging wave is a vector sum of the true reflected wave *S*11*a*1 and parasitic waves stemming from all the other sources:

( 15 ) 

The apparent reflection coefficient, perceived by the autotuner, is

( 16 ) 

where *S*11 is the true reflection coefficient and

( 17 ) 

is a parasitic term, given as a vector sum of contributions from the other feeding ports. This is the term that impairs correct tuning, and therefore should be suppressed as much as possible. The phases *ϕ*1*i* are undefined and must be assumed to be anywhere between 0° and 360°.

The worst case occurs when all the individual contributions are aligned. Then the magnitude of *δ* is

( 18 ) 

and

( 19 ) 

The phase *ϕ*1 can be anywhere between 0 and 360°.

In order for tuning to not be completely disrupted, *d* should not exceed a certain allowable level *e*, i.e.,

( 20 ) 

Practically, *e* ≈ 0.05 = 50 mU. The above equation serves for the estimation of permissible levels of cross-couplings, expressed in terms of magnitudes of transmission coefficients*S*1*i*. To simplify the matter of obtaining a practical rough estimate, we shall assume that

* all feeding powers are the same,
* the magnitudes of all cross-coupling coefficients are the same, equal to |*S*21|.

Then ( 20 ) becomes

( 21 ) 

and therefore

( 22 ) 

or, in terms of isolation, we derive our final formula

( 23 ) 

where

*I*min is the minimal permissible isolation between ports

*N* > 1 is the number of feeding ports,

*e* is the maximal tolerated magnitude of the true reflection coefficient perturbation.

The values of *I*min for a number of feeding points are illustrated in Fig. 6. As an example, if *e* = 50 mU (dashed vertical line in Fig. 6), then a two-port feeding requires the isolation at least 26 dB while a four-port feeding needs at least 36 dB.



Fig. 6. Minimal permissible isolation between feeding ports for a multiport applicator feeding.

1. Motor acceleration and deceleration have been neglected. [↑](#footnote-ref-1)
2. Higher-order contributions are those containing products of *Sij*. [↑](#footnote-ref-2)