S11 Feed Specification

October 28, 2016

1 HERA Memo 19: Establishing a Specification for the Feed Efficiency

1.0.1 Author: Aaron Parsons

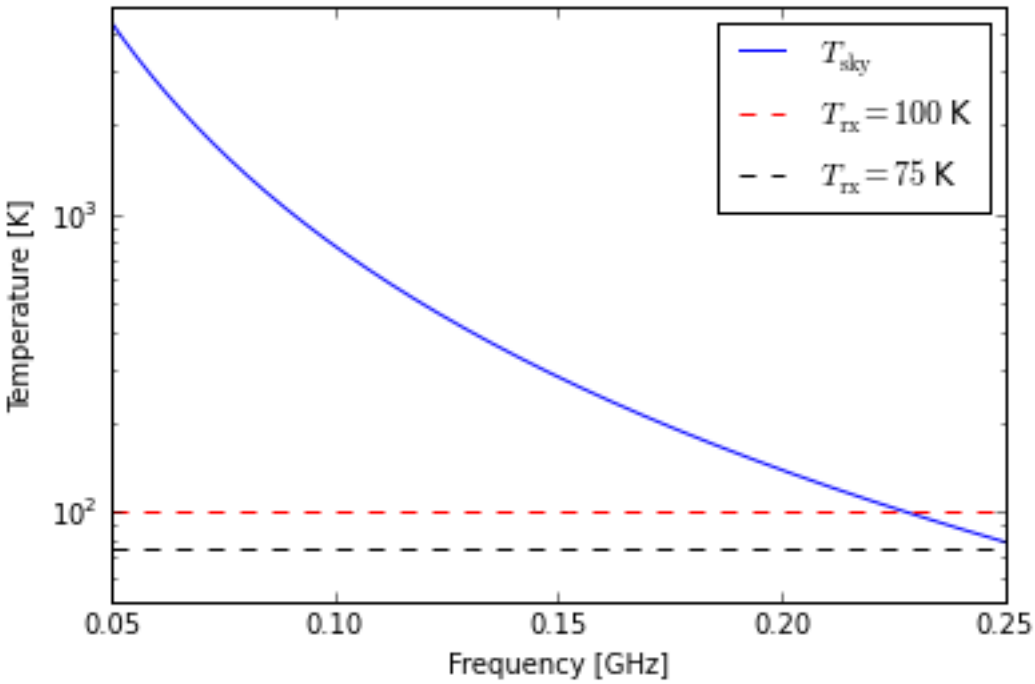
In other work (e.g. Ewall-Wice et al. 2016, Thyagarajan et al. 2016, Patra et al. 2016), we have established a specification for delay domain response of the HERA feed, which translates into a constraint on the spectral smoothness of the transmission/reflection coefficients of the feed. These constraints, while moderately sensitive to the overall magnitude of reflections off of the feed, do not actually constrain a peak amplitude of, e.g., the S_{11} coefficient for the feed. In this document, we set out to establish an overall constraint on the magnitude of S_{11} for the HERA feed on the basis of its effect on HERA's system temperature.

1.1 1. The relative temperatures of the sky and the receiver

We'll begin by assuming a sky temperature of $T_{\rm sky} \sim 180$ K at 180 MHz, corresponding approximately to a the minimum temperature in a cold patch. The spectral index of $\alpha = -2.5$ is canonical (e.g. Rogers & Bowman 2008). We adopt two receiver temperatures: $T_{\rm rx} \sim 100$ K is the official specification for the LNA/post-amp system and $T_{\rm rx} \sim 75$ K the approximate receiver temperature we might expect (private communication, Nima Razavi, 2016).

```
In [87]: import numpy as np
         %matplotlib inline
         import pylab as plt

         def calc_Tsky(nu, fq0=.180, T0=180., alpha=-2.5):
             '''Tsky = T0 * (nu/fq0)^alpha.  nu, fq0 in GHz, T0 in K.'''
             return T0 * (nu/fq0)**alpha

         nu = np.linspace(.050,.250, 1024)
         Trx1 = 100 # K
         Trx2 = 75 # K
         Tsky = calc_Tsky(nu)

         plt.semilogy(nu, Tsky, 'b', label='$T_{\\rm sky}$')
         plt.semilogy(nu, Trx1 * np.ones_like(nu), 'r--', label='$T_{\\rm rx}=100$
         plt.semilogy(nu, Trx2 * np.ones_like(nu), 'k--', label='$T_{\\rm rx}=75$ K
         plt.xlim(.05,.25); plt.ylim(50, 5000)
```

```
plt.xlabel('Frequency [GHz]')
plt.ylabel('Temperature [K]')
plt.legend(loc='best')
plt.show()
```



1.2 2. A target system temperature relative to forecasts

The impact of system temperature is to increase the amount of integration time required to achieve a given sensitivity, per the radiometer equation:

$$\sigma_{\rm rms} = \frac{T_{\rm sys}}{\sqrt{2 \cdot B \cdot t}}. \tag{1}$$

Our sensitivity forecasts for HERA were based assuming $T_{\rm sys} = 500$ K at 150 MHz, where $T_{\rm sky} \sim 280$ K. To achieve this system temperature, we need a reciever temperature that satisfies

$$T_{\rm sys} = T_{\rm sky} + \frac{1}{\epsilon} T_{\rm rx}, \tag{2}$$

where ϵ is the antenna efficiency. The parameter ϵ includes the efficiency of the dish and feed. Assuming a dish efficiency of $\epsilon_{\rm dish} = 0.74$ (HERA Memo 5; DeBoer & Parsons 2015), we obtain a target effective receiver temperature at 150 MHz of

```
In [64]: e_dish = 0.74
         eT_rx = np.around(e_dish * (500 - calc_Tsky(.150)))
         print 'T_rx,eff =', eT_rx, 'K'
```

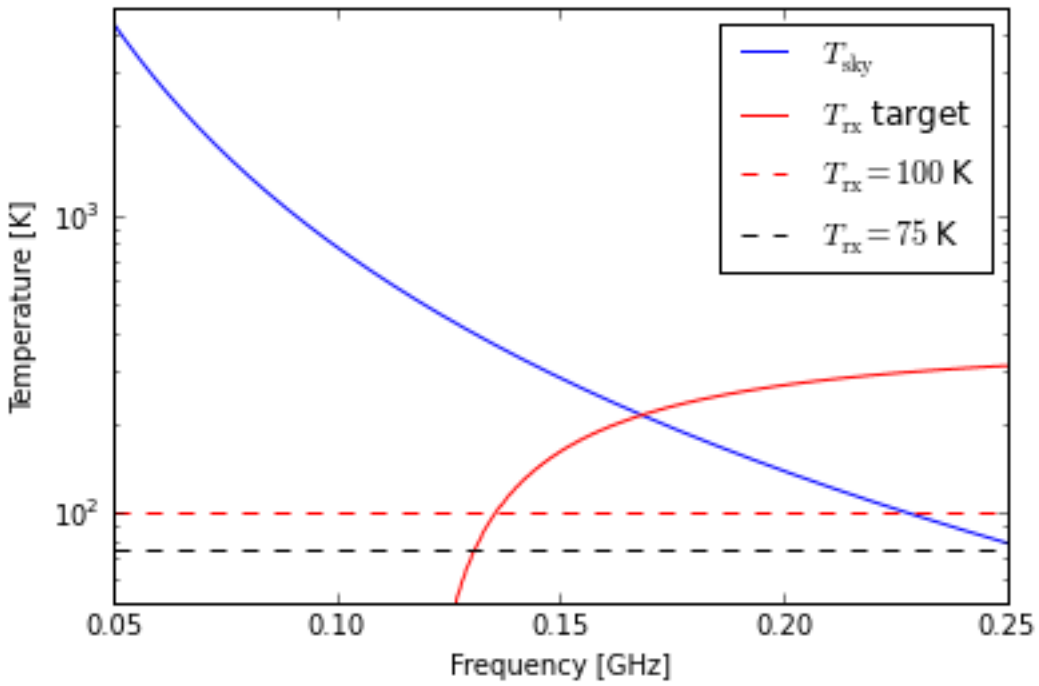
```
T_rx,eff = 160.0 K
```

Relative to our fiducial actual (not effective) receiver temperatures of $T_{\rm rx} = 100$ K and 75 K, this implies that the transmission coefficient into the feed cannot be less than $T_{\rm rx}/160$ K, corresponding to

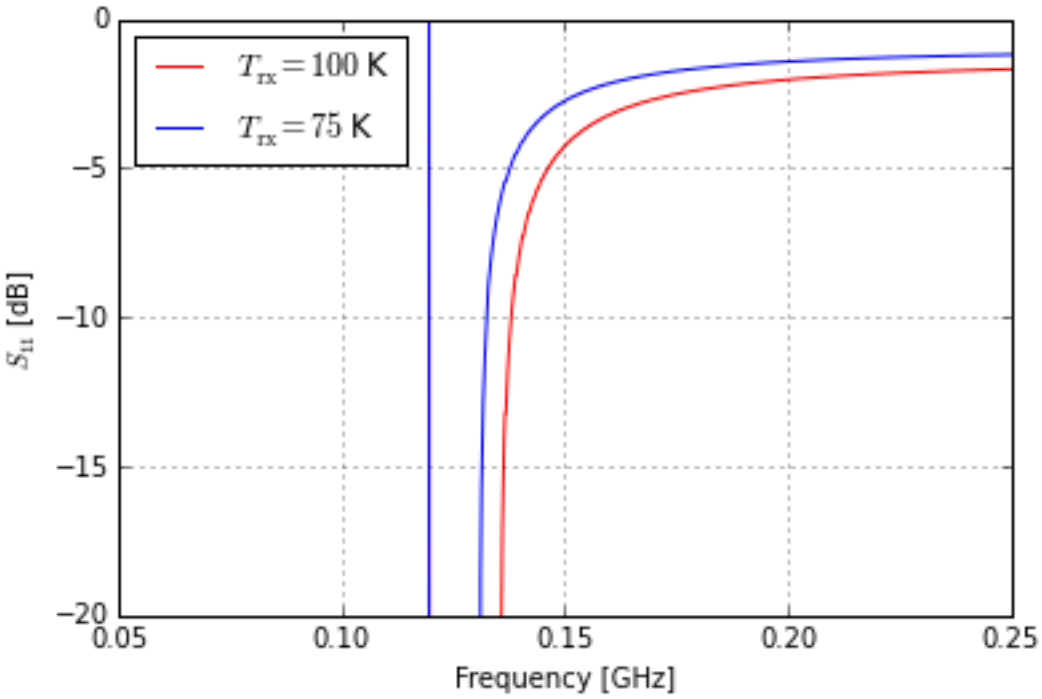
```
In [65]: S11_150_1 = np.around(10*np.log10(1 - Trx1/eT_rx))
         S11_150_2 = np.around(10*np.log10(1 - Trx2/eT_rx))
         print 'S_11 =', S11_150_1, 'dB and', S11_150_2,'dB, respectively.'

S_11 = -4.0 dB and -3.0 dB, respectively.
```

This turns out to not be a very stringent requirement at 150 MHz (we were conservative with our sensitivity estimates). This constraint is illustrated below as a function of frequency.

```
In [80]: eT_rx = np.around(e_dish * (500 - Tsky))
         plt.semilogy(nu, Tsky, 'b', label='$T_{\\rm sky}$')
         plt.semilogy(nu, eT_rx, 'r', label='$T_{\\rm rx}$ target')
         plt.semilogy(nu, Trx1 * np.ones_like(nu), 'r--', label='$T_{\\rm rx}=100$
         plt.semilogy(nu, Trx2 * np.ones_like(nu), 'k--', label='$T_{\\rm rx}=75$ K
         plt.xlim(.05,.25); plt.ylim(50, 5000)
         plt.xlabel('Frequency [GHz]')
         plt.ylabel('Temperature [K]')
         plt.legend(loc='best')
         plt.show()
```



```
In [75]: S11_1 = (1 - Trx1/eT_rx).clip(1e-10,1)
         S11_2 = (1 - Trx2/eT_rx).clip(1e-10,1)
         plt.plot(nu, 10*np.log10(S11_1), 'r', label='$T_{\\rm rx}=100$ K')
         plt.plot(nu, 10*np.log10(S11_2), 'b', label='$T_{\\rm rx}=75$ K')
         plt.xlim(.05,.25); plt.ylim(-20, 0)
         plt.xlabel('Frequency [GHz]')
         plt.ylabel('$S_{11}$ [dB]')
         plt.legend(loc='best')
         plt.grid()
         plt.show()
```



As we see, even though this is not a stringent constraint at 150 MHz, it rapidly becomes impossible at lower frequencies, when the temperature of the sky itself goes above 500 K.

1.3 3. A target integration time

If we wanted a constraint that extrapolates to lower frequencies, we could impose a requirement on $T_{\rm sys}$ relative to $T_{\rm sky}$ by specifying the fractional increase in observing time, f, that we are willing to accommodate relative to what is imposed by the sky:

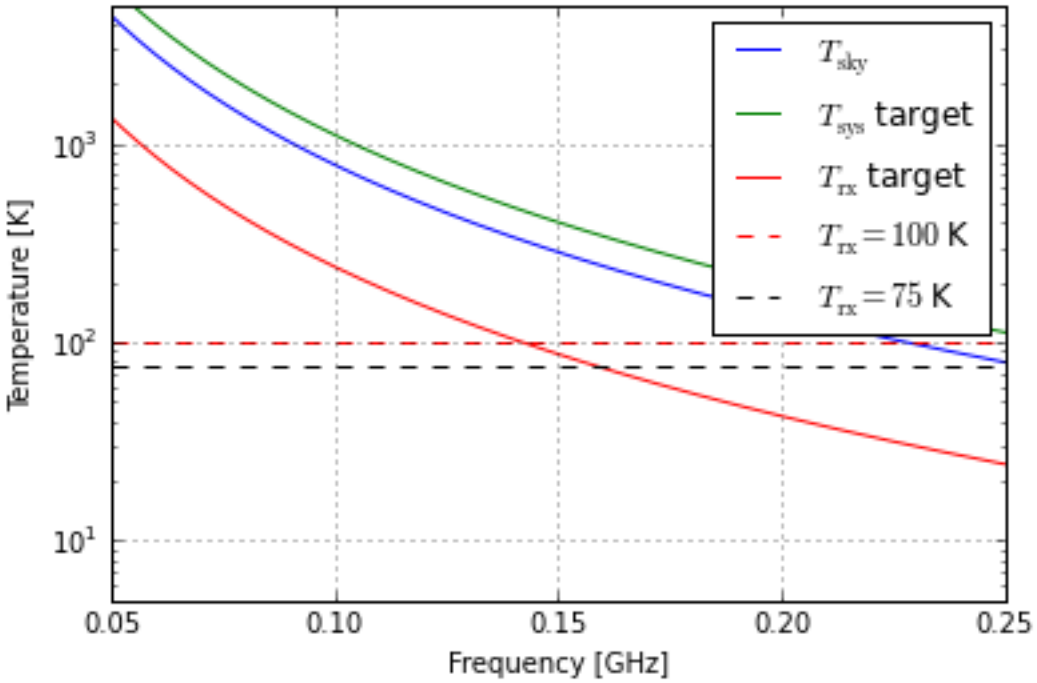
$$f = \frac{t}{t_0} = \left(\frac{T_{\rm sys}}{T_{\rm sky}}\right)^2. \tag{3}$$

For a measured actual receiver temperature, choosing a value for f determines a constraint on S_{11} for the feed according to the equation

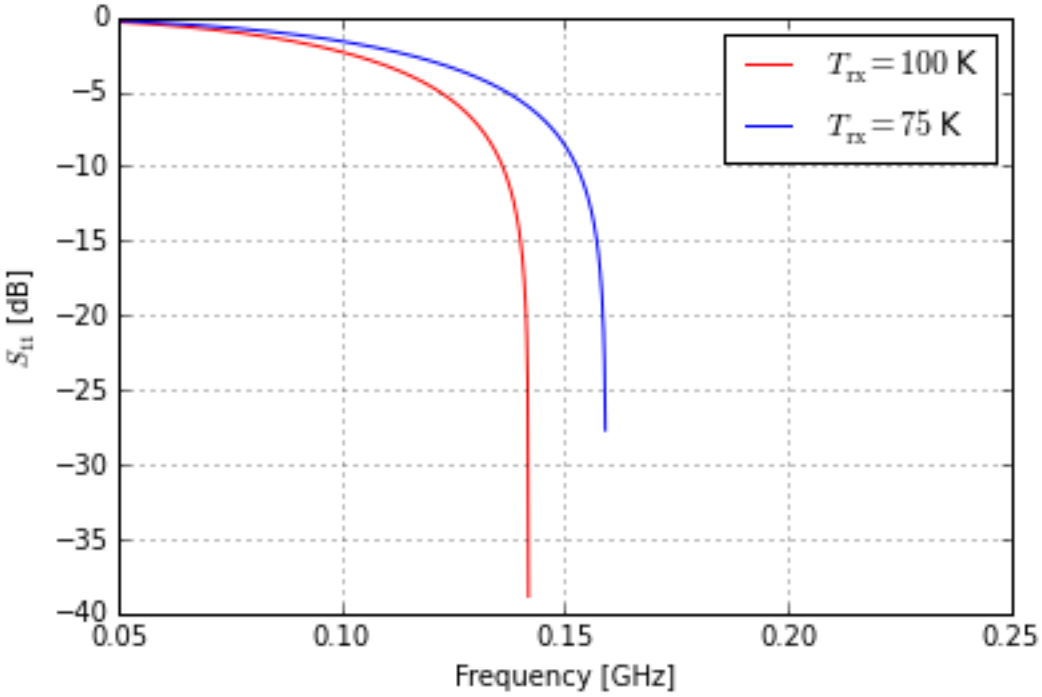
$$S_{11} = 1 - \frac{T_{\rm rx}}{T_{\rm sky}\epsilon_{\rm dish}(\sqrt{f}-1)}. \tag{4}$$

For example, choosing $f = 2$ (an observing time that is double the minimum) produces the constraint on target effective receiver temperature below in red.

```
In [71]: f_obs = 2
         e_dish = 0.74
         Tsys = np.sqrt(f_obs) * Tsky
         Trx_target = e_dish * (Tsys - Tsky)
         plt.semilogy(nu, Tsky, 'b', label='$T_{\\rm sky}$')
         plt.semilogy(nu, Tsys, 'g', label='$T_{\\rm sys}$ target')
         plt.semilogy(nu, Trx_target, 'r', label='$T_{\\rm rx}$ target')
         plt.semilogy(nu, Trx1 * np.ones_like(nu), 'r--', label='$T_{\\rm rx}=100$
         plt.semilogy(nu, Trx2 * np.ones_like(nu), 'k--', label='$T_{\\rm rx}=75$ K
         plt.xlim(.05,.25); plt.ylim(5, 5000)
         plt.xlabel('Frequency [GHz]')
         plt.ylabel('Temperature [K]')
         plt.legend(loc='best')
         plt.grid()
         plt.show()
```



```
In [77]: S11_1 = (1 - Trx1 / Trx_target).clip(0,1)
         S11_2 = (1 - Trx2 / Trx_target).clip(0,1)
         plt.plot(nu, 10*np.log10(S11_1), 'r', label='$T_{\\rm rx}=100$ K')
         plt.plot(nu, 10*np.log10(S11_2), 'b', label='$T_{\\rm rx}=75$ K')
         plt.xlim(.05,.25)#; plt.ylim(50, 5000)
         plt.xlabel('Frequency [GHz]')
         plt.ylabel('$S_{11}$ [dB]')
         plt.legend(loc='best')
         plt.grid()
         plt.show()
```



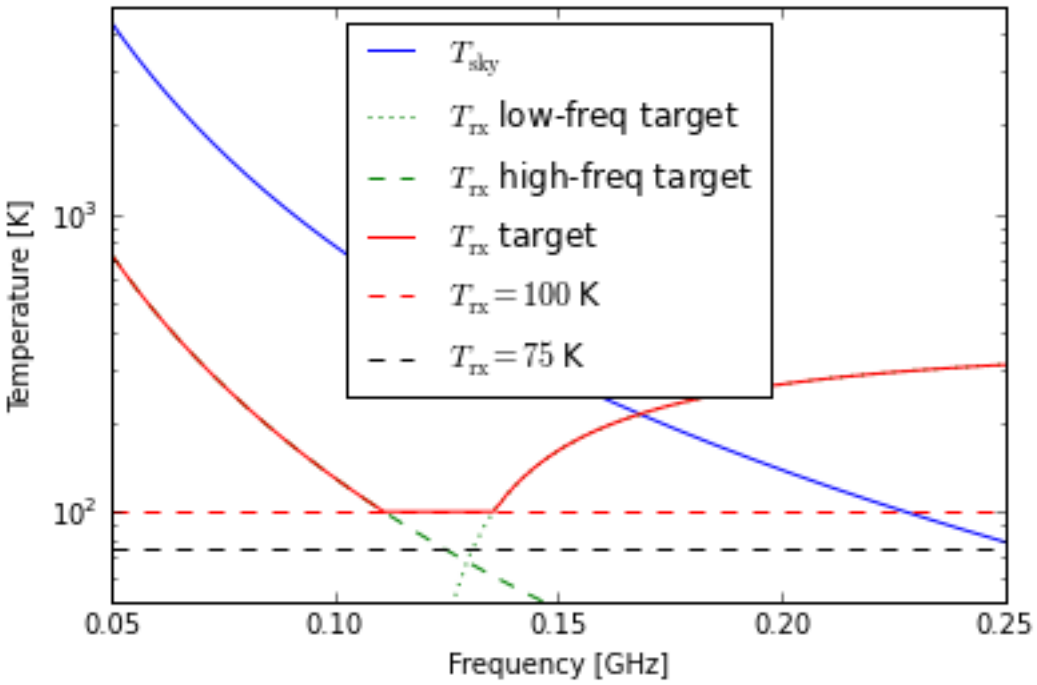
As the above figures illustrate, this specification is aggressive at 150 MHz and quickly becomes untenable at higher frequencies.

1.4 4. A piecewise specification for S_{11}

In the end, it appears we have a few distict domains to consider for our specification: 1) $T_{\rm sky} \gg 500K$, where we should indicate a specification in terms of increased integration time, 2) $T_{\rm sky} \ll 500$ K, where we would like to maintain a system temperature below 500 K, and 3) near $T_{\rm sky} \sim 500K$, we have an ambiguous zone where it may be difficult to achieve either specification. We suggest an overall specification that is the maximum of three specifications at each frequency–one corresponding to $f < 1.5$, one corresponding to $T_{\rm sys} < 500$ K, and one corresponding to $T_{\rm rx} < 100$ K, as illustrated below.

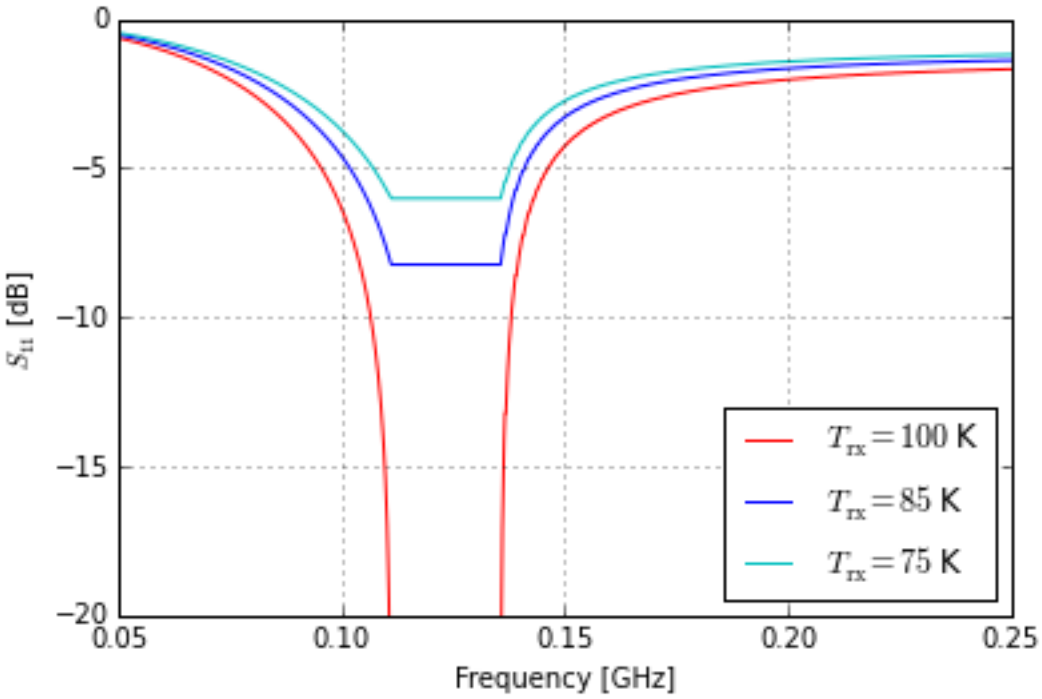
```
In [86]: f_obs = 1.5
         e_dish = 0.74
```

```
Trx_lotarg = np.around(e_dish * (500 - Tsky))
Tsys = np.sqrt(f_obs) * Tsky
Trx_hitarg = e_dish * (Tsys - Tsky)
Trx_targ = np.where(Trx_lotarg > Trx_hitarg, Trx_lotarg, Trx_hitarg).clip(
plt.semilogy(nu, Tsky, 'b', label='$T_{\\rm sky}$')
plt.semilogy(nu, Trx_lotarg, 'g:', label='$T_{\\rm rx}$ low-freq target')
plt.semilogy(nu, Trx_hitarg, 'g--', label='$T_{\\rm rx}$ high-freq target'
plt.semilogy(nu, Trx_targ, 'r', label='$T_{\\rm rx}$ target')
plt.semilogy(nu, Trx1 * np.ones_like(nu), 'r--', label='$T_{\\rm rx}=100$
plt.semilogy(nu, Trx2 * np.ones_like(nu), 'k--', label='$T_{\\rm rx}=75$ K
plt.xlim(.05,.25); plt.ylim(50, 5000)
plt.xlabel('Frequency [GHz]')
plt.ylabel('Temperature [K]')
plt.legend(loc='best')
plt.show()
```



In [92]:
```
S11_1 = (1 - Trx1 / Trx_targ).clip(0,1)
S11_2 = (1 - Trx2 / Trx_targ).clip(0,1)
S11_3 = (1 - 85 / Trx_targ).clip(0,1)
plt.plot(nu, 10*np.log10(S11_1), 'r', label='$T_{\\rm rx}=100$ K')
plt.plot(nu, 10*np.log10(S11_3), 'b', label='$T_{\\rm rx}=85$ K')
plt.plot(nu, 10*np.log10(S11_2), 'c', label='$T_{\\rm rx}=75$ K')
plt.xlim(.05,.25); plt.ylim(-20, 0)
plt.xlabel('Frequency [GHz]')
plt.ylabel('$S_{11}$ [dB]')
```

```
plt.legend(loc='best')
plt.grid()
plt.show()
```

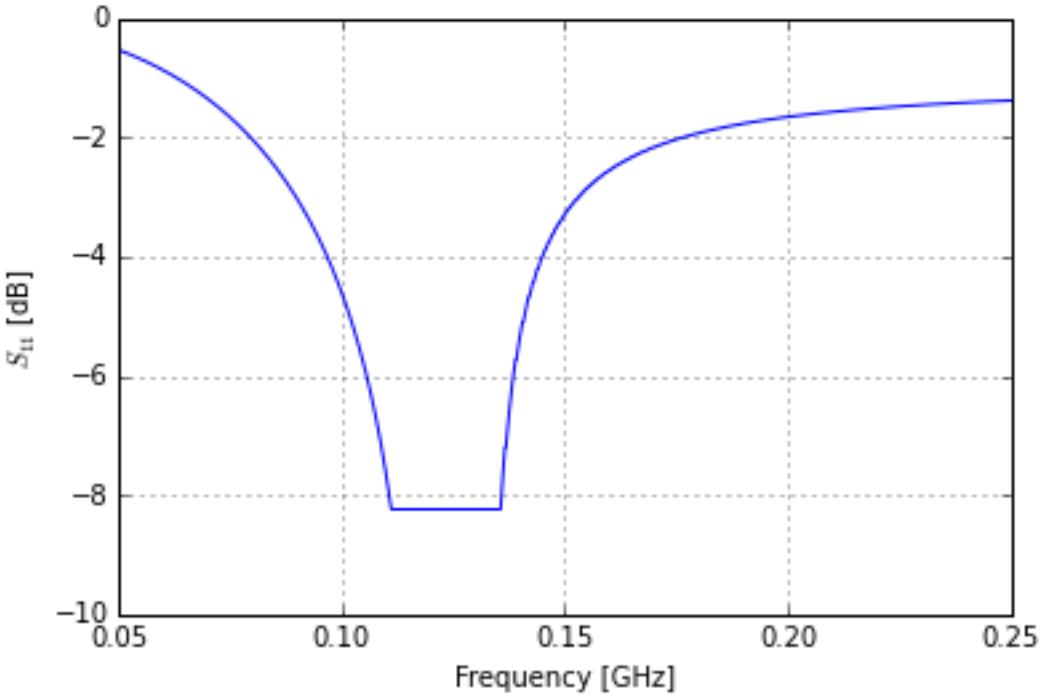


In order to meet the overall specification for system temperature, we therefore suggest adopting a specification of $T_{\rm rx} < 85$ K for the receiver system in isolation and a constraint on S_{11} for the feed corresponding to the blue curve above.

2 5. Summary

We suggest adopting a specification of $T_{\rm rx} < 85$ K for the receiver system in isolation and the constraint on S_{11} for the feed system to be below the curve illustrated below.

```
In [93]: plt.plot(nu, 10*np.log10(S11_3), 'b', label='$T_{\\rm rx}=85$ K')
         plt.xlim(.05,.25); plt.ylim(-10, 0)
         plt.xlabel('Frequency [GHz]')
         plt.ylabel('$S_{11}$ [dB]')
         plt.grid()
         plt.show()
```


0
−2
−4
−6
−8
−10
S_{11} [dB]
0.05
0.10
0.15
0.20
0.25
Frequency [GHz]