BJT Problems and Padre Exercise

Dragica Vasileska and Gerhard Klimeck
(ASU, Purdue)

1. (a) Start from the general definition of the *h*-parameters in the common-base configuration and derive the corresponding ones in the common-emitter configuration. In your derivations first relate the parameters h_{11}, h_{12}, h_{21} and h_{22} from Figure 1(a) to the ones shown in Figure 1(b) for the case when the base spreading resistance $r_{bb'} = 0$.

   (b) How will the *h*-parameters in both configurations change with the addition of the base spreading resistance $r_{bb'}$? You can arrive at approximate expressions as well considering the smallness of some of the *h*-parameters.

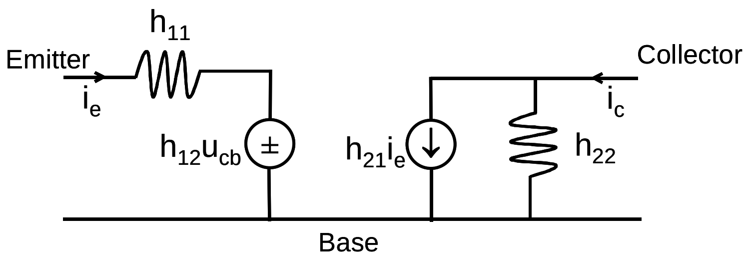


Figure 1(a)

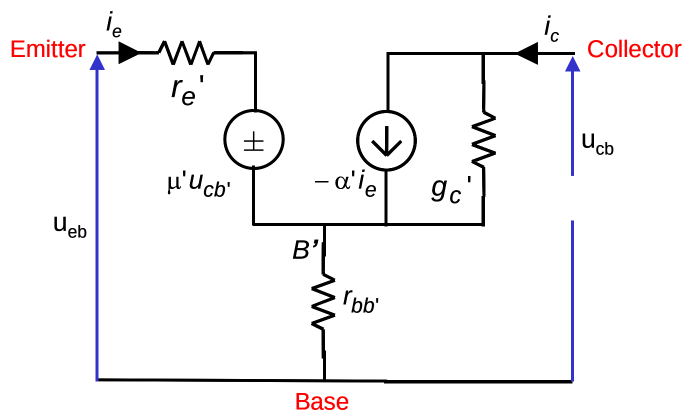
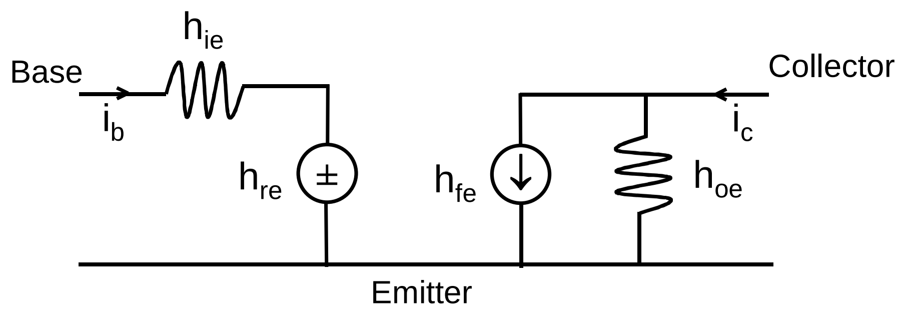


Figure 1(b)

2. Convert the input decks written for Silvaco Software to be run by Padre. Calculate the Gummel plot, the output characteristics and the h-parameters. You are given two sets of input decks. The first input deck calculates the Gummel plot and the output characteristics. The second input deck calculates the curves needed to extract the h-parameters in common emitter configuration. At low frequencies, the small-signal representation of a BJT in the common-emitter configuration is the following:



The *h* - (hybrid) parameters, that appear in this circuit, are found from the relationship of the input and output voltages and currents in this two-port network:

$$v_{be} = h_{ie} i_b + h_{re} v_{ce}$$
$$i_c = h_{fe} i_b + h_{oe} v_{ce}$$

This description leads to the following definition of the *h*-parameters:

$$h_{ie} = \left. \frac{v_{be}}{i_b} \right|_{v_{ce}=0} = \left. \frac{\partial V_{BE}}{\partial I_B} \right|_{V_{CE}=const.} \quad , \quad h_{fe} = \left. \frac{i_c}{i_b} \right|_{v_{ce}=0} \left. \frac{\partial I_C}{\partial I_B} \right|_{V_{CE}=const.}$$

$$h_{re} = \left. \frac{v_{be}}{v_{ce}} \right|_{i_b=0} = \left. \frac{\partial V_{BE}}{\partial V_{CE}} \right|_{I_B=const.} \quad , \quad hoe = \left. \frac{i_c}{v_{ce}} \right|_{i_b=0} = \left. \frac{\partial I_C}{\partial V_{CE}} \right|_{I_B=const.}$$

One can use the above definitions of the *h*-parameters to generate the proper DC curves from which you can extract all four small signal parameters.

Input Deck 1:

```
go atlas
TITLE Bipolar Gummel plot and  IC/VCE with constant IB
# Silvaco International 1992, 1993, 1994

mesh
x.m l=0   spacing=0.15
x.m l=0.8 spacing=0.15
x.m l=1.0 spacing=0.03
x.m l=1.5 spacing=0.12
x.m l=2.0 spacing=0.15

y.m l=0.0  spacing=0.006
y.m l=0.04 spacing=0.006
y.m l=0.06 spacing=0.005
y.m l=0.15 spacing=0.02
y.m l=0.30 spacing=0.02
y.m l=1.0  spacing=0.12

region num=1 silicon

electrode num=1 name=emitter left length=0.8
electrode num=2 name=base    right length=0.5 y.max=0
electrode num=3 name=collector bottom

doping reg=1 uniform n.type conc=5e15
doping reg=1 gauss   n.type conc=1e18 peak=1.0 char=0.2
doping reg=1 gauss   p.type conc=1e18 peak=0.05 junct=0.15
doping reg=1 gauss   n.type conc=5e19 peak=0.0  junct=0.05 x.right=0.8
doping reg=1 gauss   p.type conc=5e19 peak=0.0  char=0.08 x.left=1.5

# set bipolar models
```

```
models conmob fldmob consrh auger print
contact name=emitter n.poly surf.rec

solve init

save outf=bjtex04_0.str

tonyplot  bjtex04_0.str -set bjtex04_0.set

# Gummel plot

method  newton autonr trap
solve vcollector=0.025
solve vcollector=0.1
solve vcollector=0.25 vstep=0.25 vfinal=2 name=collector

solve vbase=0.025
solve vbase=0.1
solve vbase=0.2

log outf=bjtex04_0.log
solve vbase=0.3 vstep=0.05 vfinal=1 name=base

tonyplot  bjtex04_0.log -set bjtex04_0_log.set

#IC/VCE with constant IB

#ramp Vb

log off
solve init
solve vbase=0.025
solve vbase=0.05

solve  vbase=0.1 vstep=0.1 vfinal=0.7 name=base

# switch to current boundary conditions

contact name=base current

# ramp IB and save solutions
solve ibase=1.e-6
save outf=bjtex04_1.str master
solve ibase=2.e-6
save outf=bjtex04_2.str master
solve ibase=3.e-6
save outf=bjtex04_3.str master
solve ibase=4.e-6
save outf=bjtex04_4.str master
solve ibase=5.e-6
save outf=bjtex04_5.str master

# load in each initial guess file and ramp VCE
load inf=bjtex04_1.str master
log outf=bjtex04_1.log
solve vcollector=0.0 vstep=0.25 vfinal=5.0 name=collector

load inf=bjtex04_2.str master
log outf=bjtex04_2.log
solve vcollector=0.0 vstep=0.25 vfinal=5.0 name=collector

load inf=bjtex04_3.str master
log outf=bjtex04_3.log
solve vcollector=0.0 vstep=0.25 vfinal=5.0 name=collector

load inf=bjtex04_4.str master
log outf=bjtex04_4.log
solve vcollector=0.0 vstep=0.25 vfinal=5.0 name=collector

load inf=bjtex04_5.str master
log outf=bjtex04_5.log
```

```
solve vcollector=0.0 vstep=0.25 vfinal=5.0 name=collector

# plot results
tonyplot -overlay  bjtex04_1.log bjtex04_2.log bjtex04_3.log bjtex04_4.log bjtex04_5.log
-set bjtex04_1_log.set
quit
```

Input Deck 2:

```
###################################################
# This code provides solution to HW#6 - Spring 2001
# (1) calculates device output characteistics
# (2) calculates the h-parameters using the appro-
#     priate current or voltage sweeps
# (3) in this case, hfe=beta is found from slope
# (4) Does not include Gummel plot calculation
# (5) Notice the different EXTRACT statements here
###################################################

go atlas

TITLE structure of a bipolar junction transistor

###################################################
# Initial device structure and models specification
###################################################

Mesh
x.m l=0  spacing=0.15
x.m l=0.8 spacing=0.15
x.m l=1.0  spacing=0.03
x.m l=1.5 spacing=0.12
x.m l=2.0 spacing=0.15

y.m l=0.0 spacing=0.006
y.m l=0.04 spacing=0.006
y.m l=0.06 spacing=0.005
y.m l=0.15 spacing=0.02
y.m l=0.30 spacing=0.02
y.m l=1.0 spacing=0.12

region num=1 silicon

electrode num=1 name=emitter left length=0.8
electrode num=2 name=base right length=0.5 y.max=0
electrode num=3 name=collector bottom

doping reg=1 uniform n.type conc=5e15
doping reg=1 gauss n.type conc=1e18 peak=1.0 char=0.2
doping reg=1 gauss p.type conc=1e18 peak=0.05 junct=0.15
doping reg=1 gauss n.type conc=5e19 peak=0.0 junct=0.05 x.right=0.8
dop reg=1 gauss p.type conc=5e19 peak=0.0 char=0.08 x.left=1.5

# Set bipolar models
models conmob fldmob consrh auger print numcarr=2
contact name=emitter n.poly surf.rec

#############################################
# Ramping of the base voltage to obtain good
# initial guess
#############################################

# Ramp Vb
solve init
solve vbase=0.025
solve vbase=0.05
solve vbase=0.1 vstep=0.1 vfinal=0.7 name=base
save outfile=init_file.str master.in

#############################################
# Calculate hie and hfe by stepping the
```

```
# base current
# (first ramp the collector voltage to 3V)
############################################

# Ramp Vce voltage to 3V
solve vcollector=0.0 vstep=0.25 vfinal=3.0 name=collector

# Switch to current boundary conditions and ramp IB
contact name=base current
log outf=base_sweep.log
solve ibase=0.5e-6 istep=0.5e-6 ifinal=6.5e-6 name=base

# Extract the hie-parameter:
extract name="hie" grad from curve(i."base",v."base") where x.val=1.e-6
extract name="hie" grad from curve(i."base",v."base") where x.val=2.e-6
extract name="hie" grad from curve(i."base",v."base") where x.val=3.e-6
extract name="hie" grad from curve(i."base",v."base") where x.val=4.e-6
extract name="hie" grad from curve(i."base",v."base") where x.val=5.e-6
extract name="hie" grad from curve(i."base",v."base") where x.val=6.e-6

# Extract the hfe-parameter:
extract name="hfe" grad from curve(i."base",i."collector") where x.val=1.e-6
extract name="hfe" grad from curve(i."base",i."collector") where x.val=2.e-6
extract name="hfe" grad from curve(i."base",i."collector") where x.val=3.e-6
extract name="hfe" grad from curve(i."base",i."collector") where x.val=4.e-6
extract name="hfe" grad from curve(i."base",i."collector") where x.val=5.e-6
extract name="hfe" grad from curve(i."base",i."collector") where x.val=6.e-6
log off

############################################
# Calculate hre and hoe by first stepping the
# base current and then sweeping the voltage
# Vce - not very efficient method (not fast)
############################################

# Load in previously saved file:
load inf=init_file.str master

# Switch to current boundary conditions and ramp IB
contact name=base current

solve ibase=1e-6
save outf=bjt_1.str master
solve ibase=2e-6
save outf=bjt_2.str master
solve ibase=3e-6
save outf=bjt_3.str master
solve ibase=4e-6
save outf=bjt_4.str master
solve ibase=5e-6
save outf=bjt_5.str master
solve ibase=6e-6
save outf=bjt_6.str master
solve ibase=7e-6
save outf=bjt_7.str master

# Load in each initial guess file and ramp VCE
load inf=bjt_1.str master
log outf=bjt_1.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_2.str master
log outf=bjt_2.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from slope(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_3.str master
log outf=bjt_3.log
```

```
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_4.str master
log outf=bjt_4.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_5.str master
log outf=bjt_5.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_6.str master
log outf=bjt_6.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

load inf=bjt_7.str master
log outf=bjt_7.log
solve vcollector=0.0 vstep=0.25 vfinal=3.25 name=collector
extract name="hre" grad from curve(v."collector",v."base") where x.val=3.0
extract name="hoe" grad from curve(v."collector",i."collector") where x.val=3.0

quit
```

Results that one should get:

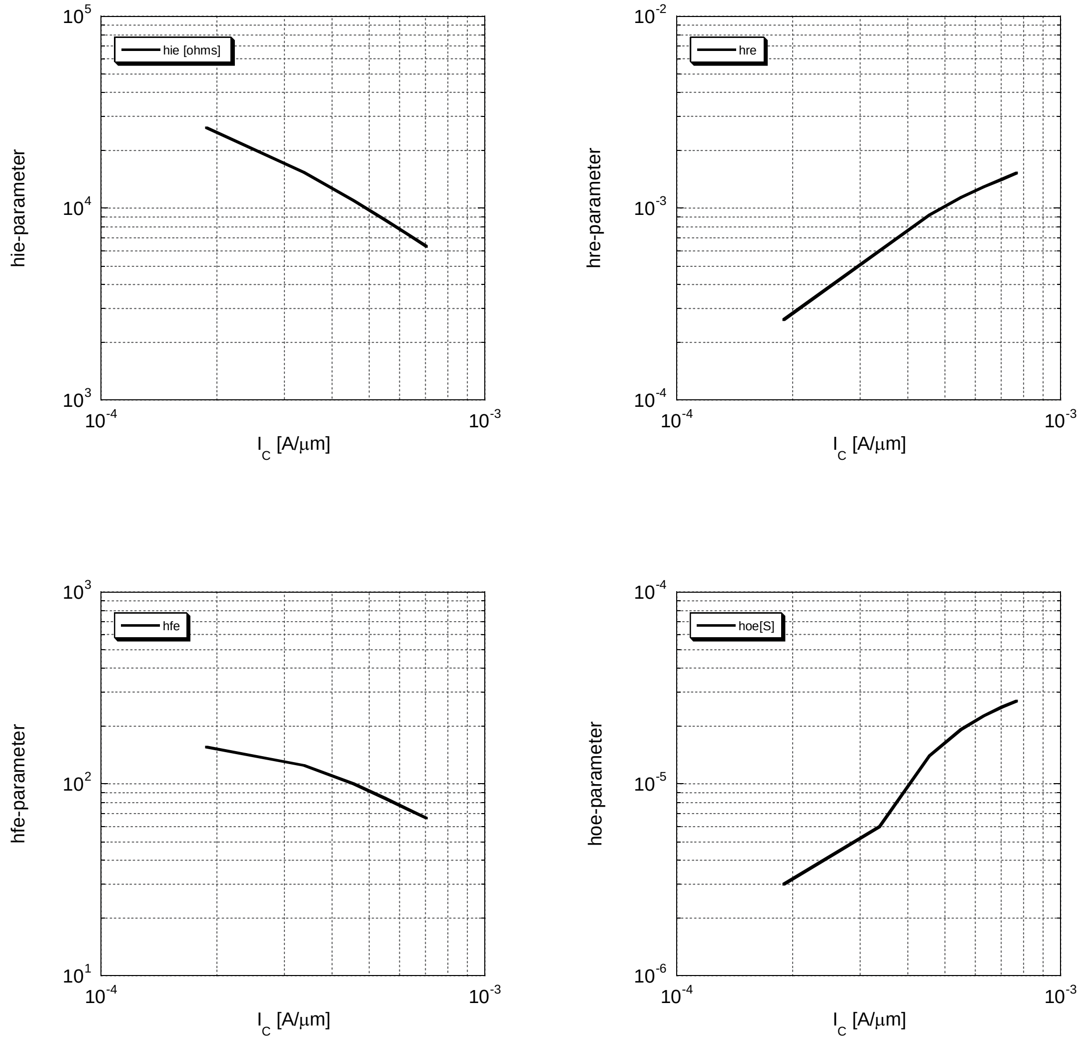
The h-parameters presented on these graphs were calculated in the following manner:

(1) For VCE=3 V, a sweep in the base current IB was done and the parameters hie and hfe were calculated using:

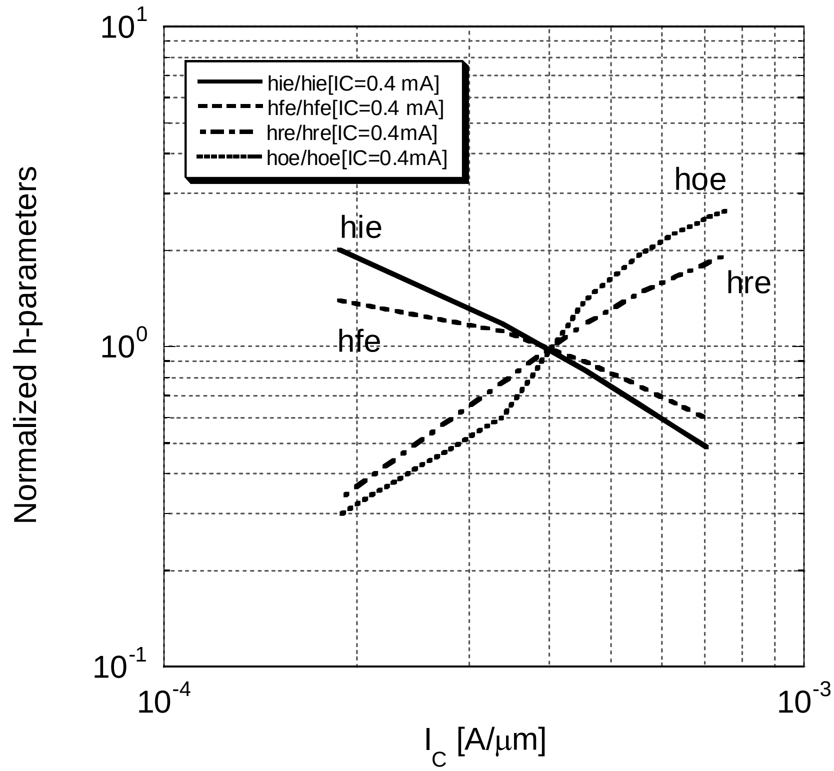
$$hie = \left.\frac{\partial V_{BE}}{\partial I_B}\right|_{V_{CE}=const.} \quad \text{and} \quad hfe = \left.\frac{\partial I_C}{\partial I_B}\right|_{V_{CE}=const.}$$

(2) For given IB, a sweep in the collector current was made and the hre and hoe parameters were calculated using:

$$hre = \left.\frac{\partial V_{BE}}{\partial V_{CE}}\right|_{I_B=const.} \quad \text{and} \quad hoe = \left.\frac{\partial I_C}{\partial V_{CE}}\right|_{I_B=const.}$$



The normalized h-parameters in the common-emitter configuration are shown in the figure below:



3. (a) For the description of two-port networks, instead of using the *S*-parameter set, one can also use the so-called *T*-parameter set that relates the input to the output variables (*T* stands for transmission). Given the following general definitions for the *S*- and *T*-parameters of a two-port network, express the elements of the *T*-matrix in terms of the elements of the *S*-matrix.

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; \quad \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} b_2 \\ a_2 \end{bmatrix}$$

(b) In most cases it is lot easier to measure the *S*-parameters under conjugate matching conditions, which do not require short or open-circuits, and calculate the *Y*-parameters. Therefore, the relationship between the two is an important one. Here you are asked the opposite problem, i.e. given the *Y*-parameters, you need to find the corresponding expressions for the elements of the *S*-matrix, i.e. to express the *S*-parameters in terms of the *Y*-parameters.