NQS effects and their implementation

Michael Schroter, Anindya Mukherjee, Anjan Chakravorty

Dept. of Electrical and Computer Engin.
University of California at San Diego
USA

CEDIC
University of Technology Dresden
Germany

Dept. of Electrical Engineering
Indian Institute of Technology Madras
Chennai 600 036, India

mschroter@ieee.org , mukherje@iee.et.tu-dresden.de , anjan@ee.iitm.ac.in
http://www.iee.et.tu-dresden.de/iee/eb/hic_new/hic_start.html

HICUM Workshop

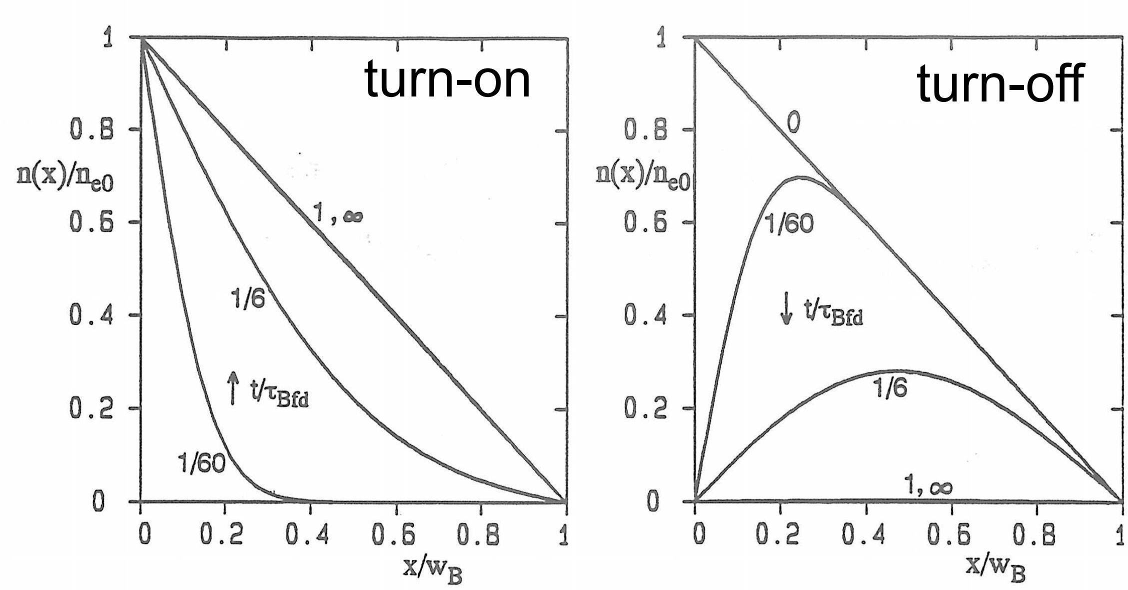
Dresden, June 18, 2007

OUTLINE

- Vertical non-quasi-static (NQS) effects: device physics
- Vertical NQS effects: compact modeling
- Vertical NQS effects: implementation
- Implementation in VA-based releases
- Results of NQS effect implementation
- Summary

Vertical non-quasi-static (NQS) effects: device physics

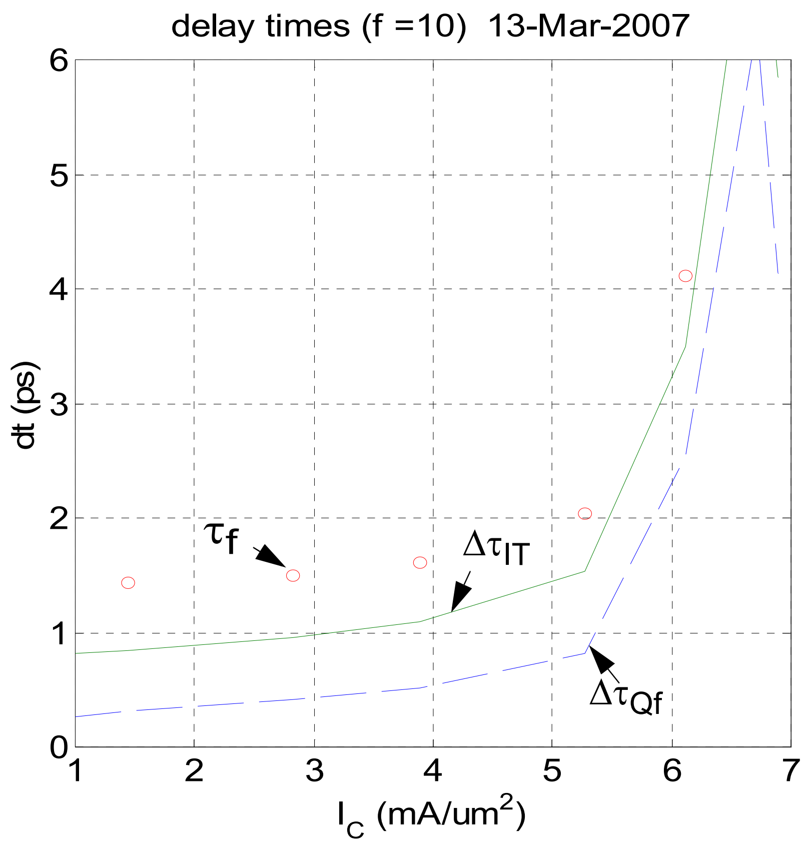
- at high-speed switching or high frequencies (h.f.) => "delayed" reaction to BE voltage change:
  - charge Q_f <u>*and*</u>
  - transfer current I_T
- visible for intrinsic transistor at about $f_T/5$

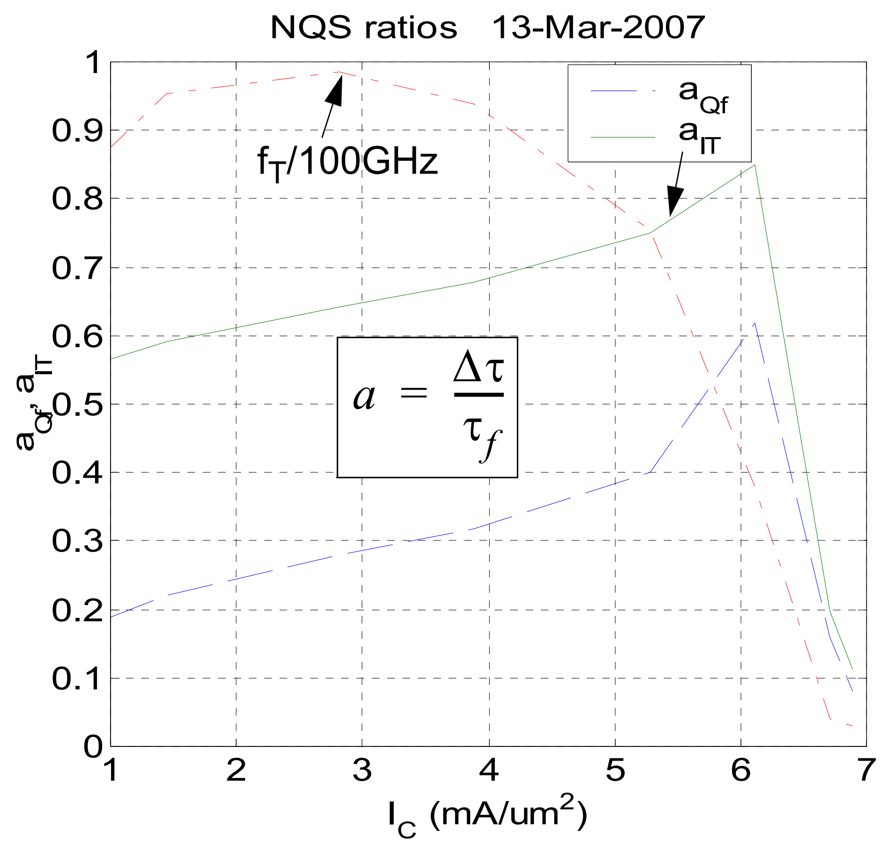


- often found: "delay approach $\exp(-j\omega\tau_d)$" for the transconductance => *small-signal based only* and no theoretical basis
- often found: charge delay ignored => incorrect current gain phase shift
- exact solution from continuity & transport equation is available but not in closed form; also, no compact generic solution for ***large-signal nonlinear*** case

SiGe HBT example

1D device simulation results





=> "delay" times are bias dependent

Vertical NQS effects: compact modeling

- frequency domain solution is truncated after first or second frequency term:

$$\underline{X}_{nqs} \approx X_{qs} - (\omega\tau_2)^2 - j\omega\tau_1 \quad (X = I_T,\ Q_f)$$

=> valid at each bias point!

- time domain solution (TICCR)
  - allows to construct arbitrarily accurate NQS description (in theory)
  - usually QS carrier distribution assumed (known in time and frequency domain)
    => partitioned-charge based (PCB) approach

$$i_C(t) = i_{T,qs} - \gamma_t \frac{\partial q_f(t)}{\partial t} - \frac{\partial q_{jC}(t)}{\partial t}$$ => additional time delay in collector current

$$i_B(t) = i_{B,qs} + \frac{\partial q_{jE}(t)}{\partial t} + \frac{\partial q_{jC}(t)}{\partial t} + \frac{\partial q_f(t)}{\partial t}$$ => no additional time delay in charge

  - theoretical evaluation yields ***small-signal*** solution identical to classical g_m

Vertical NQS effects: implementation

existing HICUM/L2 v2.1

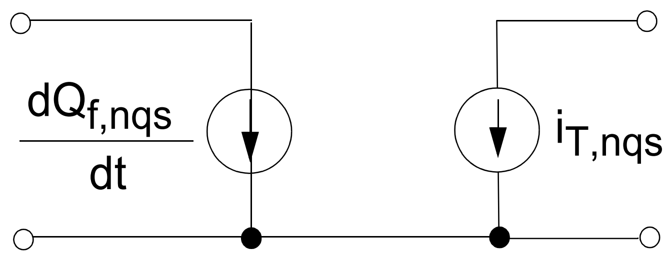
- takes into account charge *and* collector current NQS effect (ALIT, ALQF)
  => correct description of both ***current gain*** and ***g_m***

- uses Weil's approach: Bessel polynomial in frequency domain is transformed to 2nd-order differential equation in time domain

$$\underline{X}_{nqs} \approx X_{qs} - (\omega\tau_2)^2 - j\omega\tau_1 \cong \frac{X_{qs}}{1 + A_1 s + A_2 s^2} \quad \circ\!\!-\!\!\bullet \quad A_2 \frac{d^2 x_{nqs}}{dt^2} + A_1 \frac{dx_{nqs}}{dt} + x_{nqs} = x_{qs}(t)$$

- explicit form and consistent in time domain and frequency domain

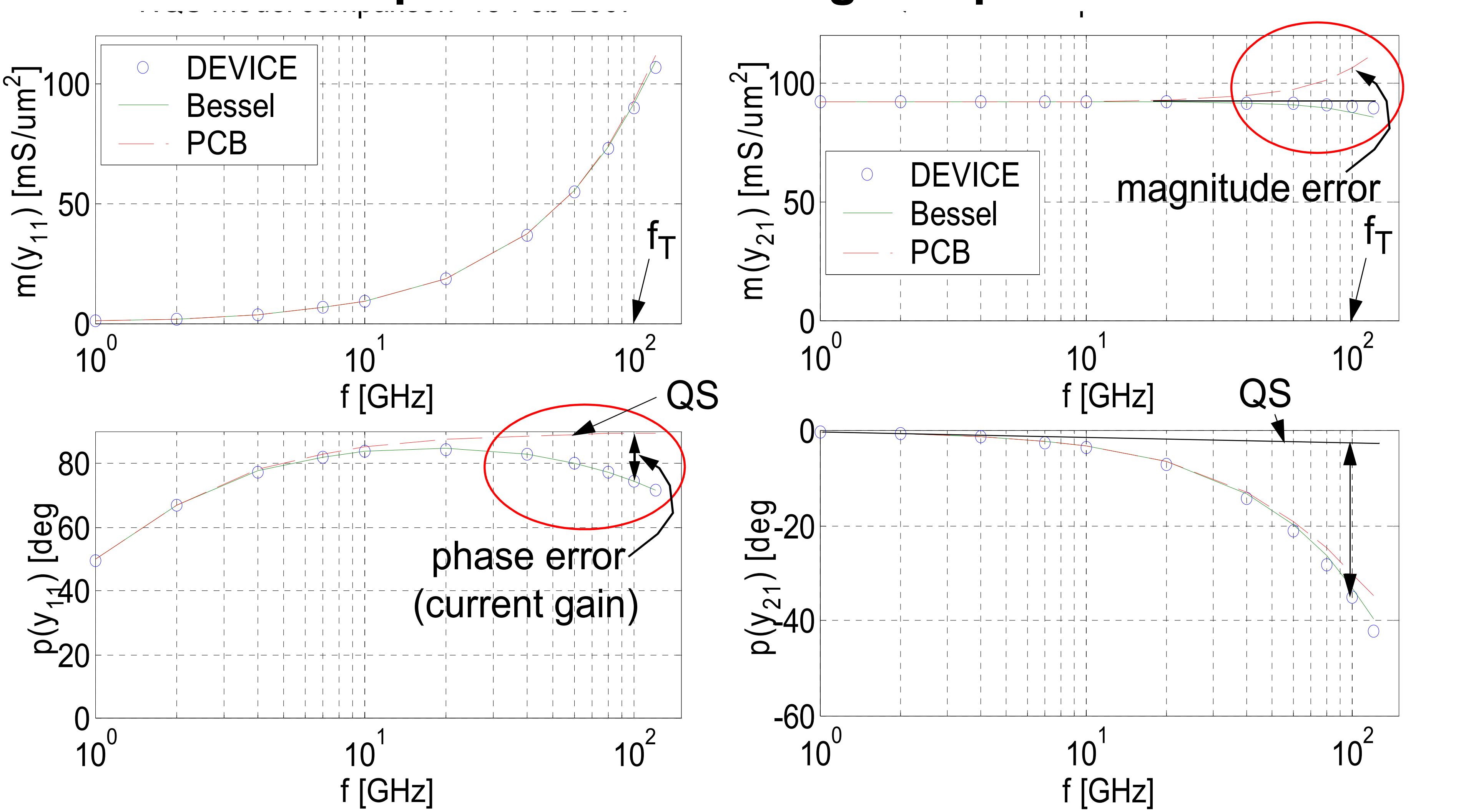
- no change in equivalent circuit



Note: simple low-pass is sufficient for charge

- Weil's approach has worked very well so far AND has indicated design errors in several known cases!!

Comparison of small-signal options



- error depends on bias and frequency (increases with current density, f)

Implementation in existing VA code

HICUM/L2 v2.21 and v2.22

- First VA-based release (without VA code for NQS effects)

- VA does not (yet?) allow to use Weil's approach directly
  => assumption: already existing Weil's code will be taken over from v2.1

- Note1: remember EDA vendor comments about VA compiler issues:
  "... need *handcrafting* in any way to optimize computational performance ..."

- Note2: release procedure seemed to be acceptable to EDA vendors (and according to existing Q&A release guidelines)

Issue:

- completely VA-based model code incl. NQS effect implementation

Implementation in VA-based releases

... and assuming new release requirement

Option1: C code only release

- generate C code from compiler
- handcraft features that are not available (yet) in VA compilers
  => code is valid for ONE specific simulator only!
- further optimization by EDA vendor

=> back to square1: simulator dependent code release

Option 2: VA code only release

- simulator independent code release (dependence is shifted to VA compiler)
- not all necessary features available (yet) in compiler: Weil’s approach, ...

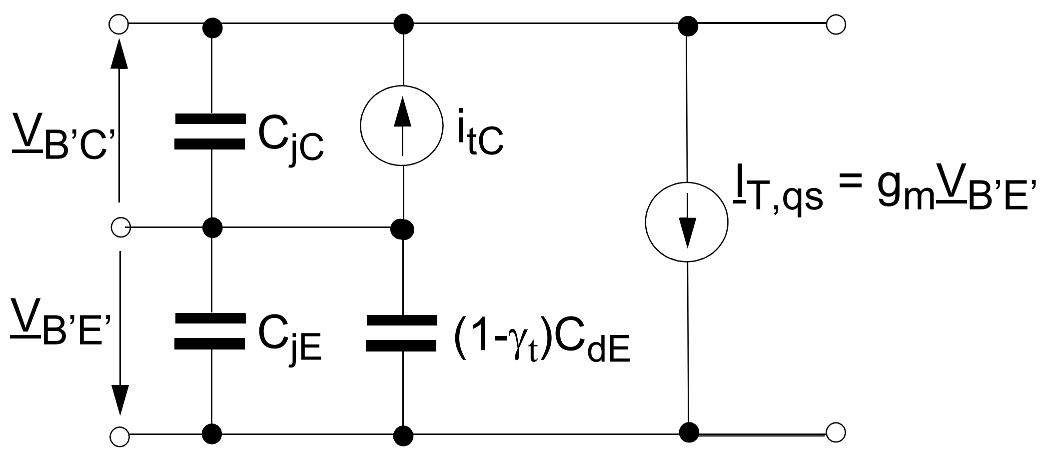
=> need other options for implementing NQS effects unless **VA construct** is made available for Weil’s approach

Implementation: VA-suitable methods (1/5)

partitioned-charge based (PCB) approach (transcapacitance)

- only phase of I_T, g_m => incorrect magnitude of y_{21} and phase of y_{11}
- diffusion charge Q_f(bias,T) and associated capacitance C_{dE}(bias,T)

=> undesired derivatives in small-signal EC are Newton/compiler artifact



assumption: γ_t = const(bias,t)

"trans"-charge: $Q_{tC} = \gamma_t Q_f(v_{B'E'}, v_{B'C})$

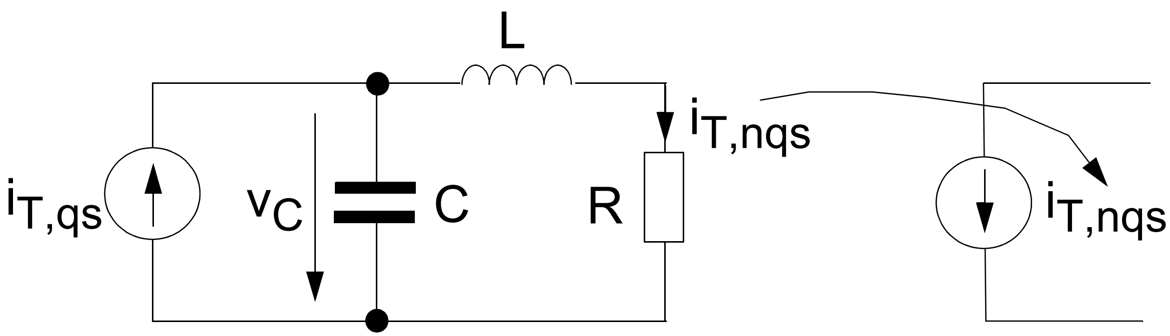
large-signal current: $i_{tC} = \dfrac{dQ_{tC}}{dt}$

small-signal current:

$$di_{tC} = \underbrace{\frac{dQ_{tC}}{dV_{B'E'}} \frac{dv_{B'E'}}{dt}}_{= \gamma_t C_{dE}} + \underbrace{\frac{dQ_{tC}}{dV_{B'C'}} \frac{dv_{B'C'}}{dt}}_{\text{undesired}}$$

Implementation options (2/5)

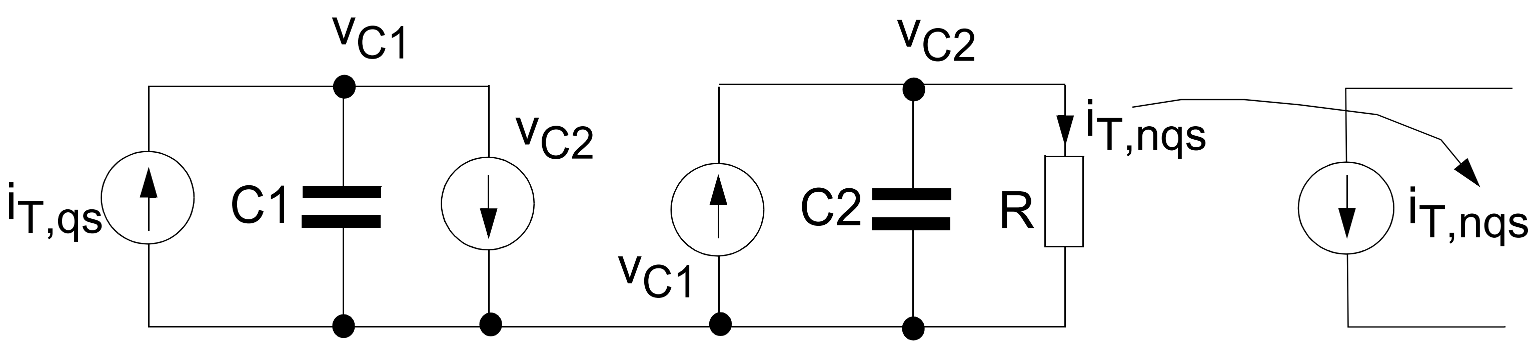
adjunct LCR network (or gyrator equivalent)



$C = C(v_{B'E'}, v_{B'C})$

$L = L(v_{B'E'}, v_{B'C})$

$R = 1\Omega$

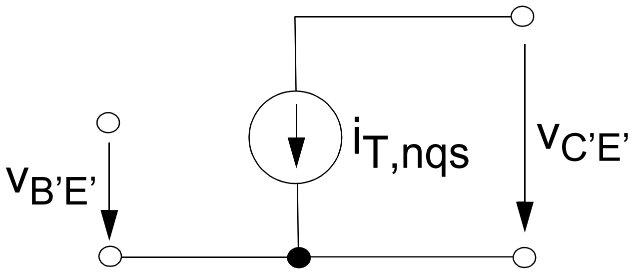


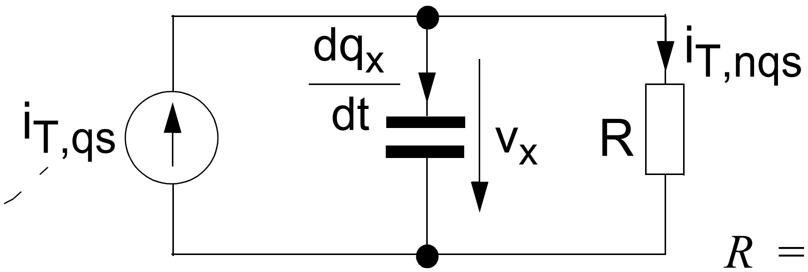
- represents 2nd-order polynomial **<u>exactly</u>** in time- **<u>and</u>** -frequency domain
- used (e.g. VBIC) so far for I_T, g_m only, but with *<u>(bias,T) independent</u>* delay time
- need to (i) make values (bias,T) dependent, (ii) add charge-related adjunct network

=> VA compiler generates undesired derivatives in s-s EC

NQS adjunct networks: implementation issues (3/5)

Example: simple NQS model for large-signal (time domain) transfer current





$$R = 1\Omega$$

$$i_{T,qs}(v_{B'E'}, v_{C'E'}) \qquad q_x = C(i_{T,qs})v_x \quad \text{with} \quad C(i_{T,qs}) = a_{IT}\tau_f(i_{T,qs})$$

generated small-signal elements ...

$$g_m = \frac{di_{T,qs}}{dv_{B'E'}} \qquad g_o = \frac{di_{T,qs}}{dv_{C'E'}}$$

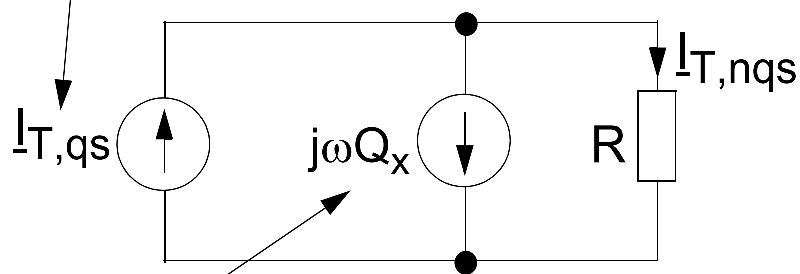
$$C = \frac{dq_x}{dv_x}$$

$$C_{xe} = \frac{dq_x}{dv_{B'E'}} = V_x\frac{dC}{dv_{B'E'}} = a_{IT}g_m\frac{d\tau_f}{di_{T,qs}}$$

$$C_{xc} = \frac{dq_x}{dv_{C'E'}} = V_x\frac{dC}{dv_{C'E'}} = a_{IT}g_o\frac{d\tau_f}{di_{T,qs}}$$

... and associated equivalent circuit

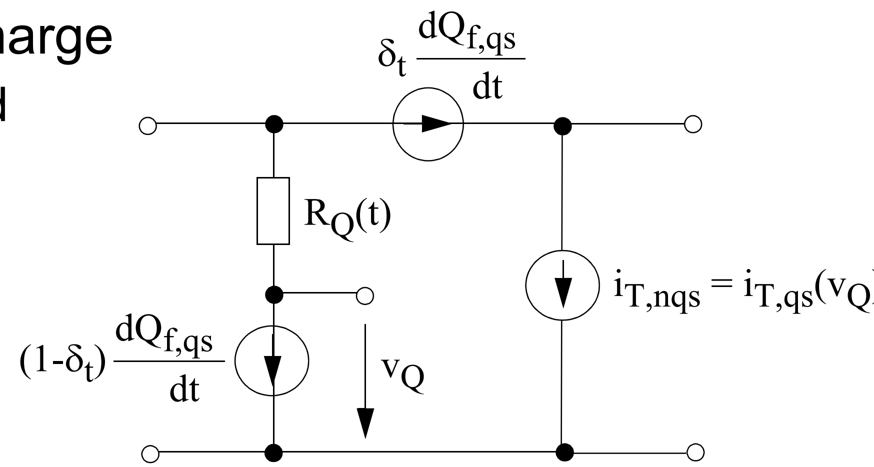
$$\Delta i_{T,qs} = g_m\Delta v_{B'E'} + \underbrace{g_o\Delta v_{C'E'}}_{\text{undesired for NQS}}$$



$$\Delta q_x = C\Delta v_x + \underbrace{C_{xe}\Delta v_{B'E'} + C_{xc}\Delta v_{C'E'}}_{\text{undesired}}$$

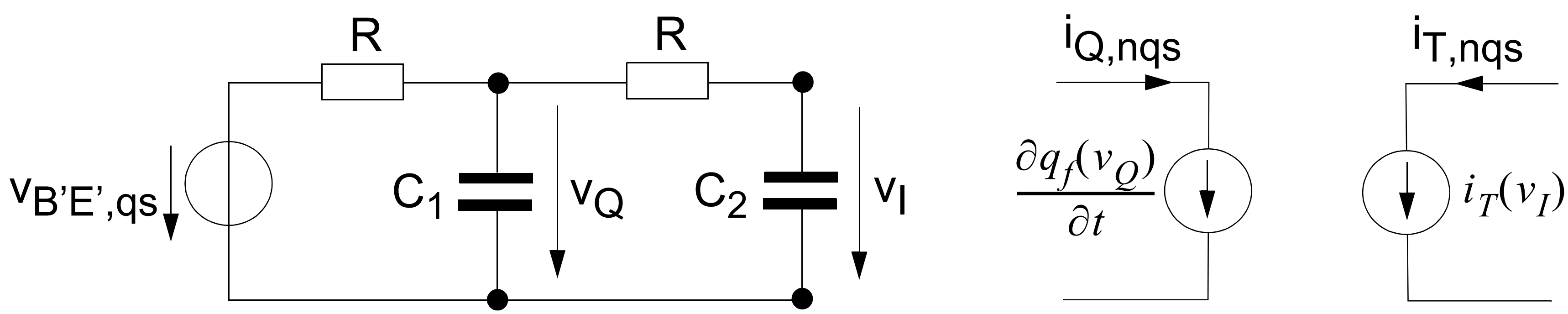
Implementation options (4/5)

- RC network instead of QS diffusion charge
  - so far used only for small-signal EC and (bias,T) independent delay time
  - R_Q is frequency dependent (ω^2)!!
  - adds permanent additional internal node



=> not practically feasible

- delay controlling voltage (delayed electron quasi-fermi-potential)
  - with different time constants for Q_f and I_T
  - cascaded low-pass



=> never tested in BJT/HBT models => physical approach?

Implementation options (5/5)

- base region discretization (collocation method)
  - generates permanent nodes
    => unacceptable increase in simulation time, especially if effect is not desired
  - note: base region is not the only relevant region with charge
  - additional issues
    - structural information (often considered as IP by foundries)
    - more or less significant "reformulation" of other model equations

=> not an attractive option

- keep delay times bias independent
  - incompatible to previous version 2.1 and implemented 2.2 versions

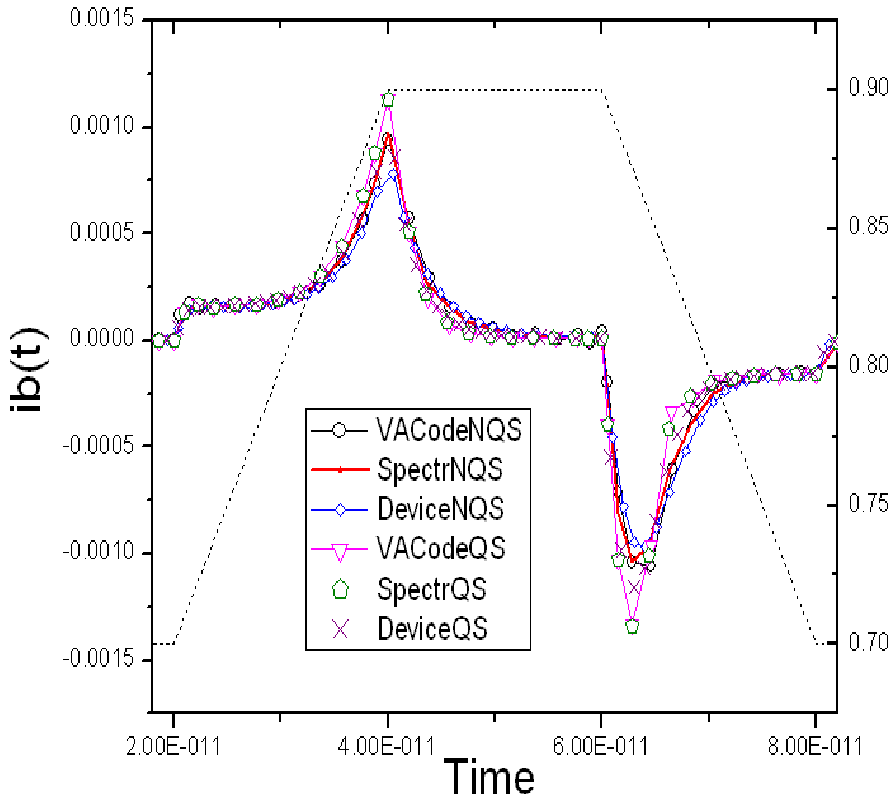
=> intermediate solution

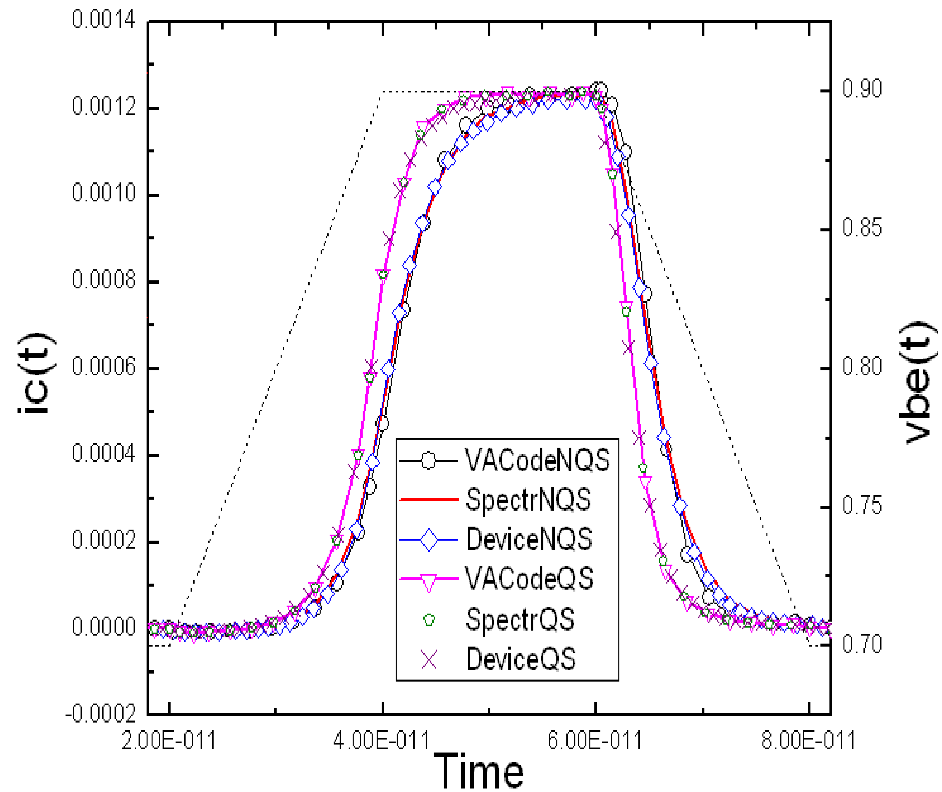
- omit NQS effects
  => incompatibility to previous versions (e.g. HICUM v2.1)
  => h.f. design impact

Results of NQS effect implementation

HICUM comparison: v2.1(Weil) vs. v2.2 (adj. netw.)

Large-signal transient simulation (first results)

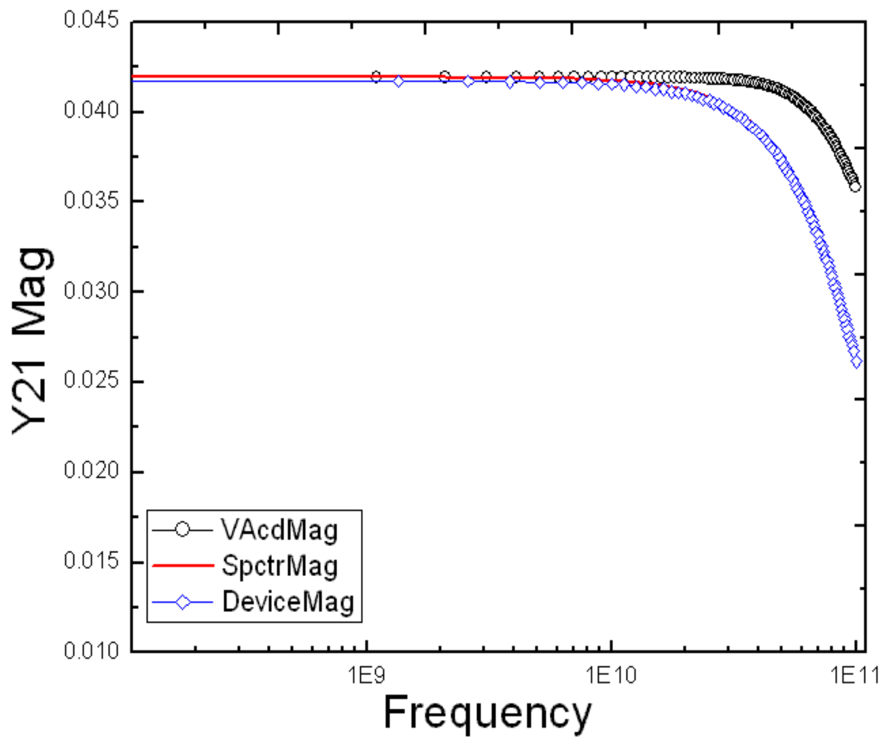


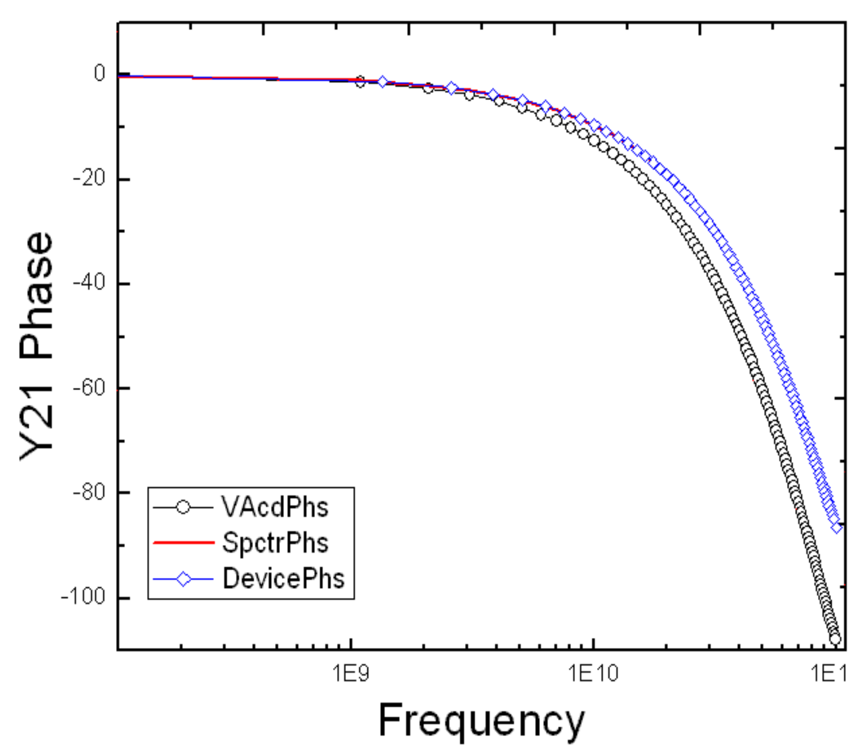


=> acceptable agreement

HICUM comparison: v2.1(Weil) vs. v2.2 (adj. netw.)

Small-signal frequency-domain simulation: Y21



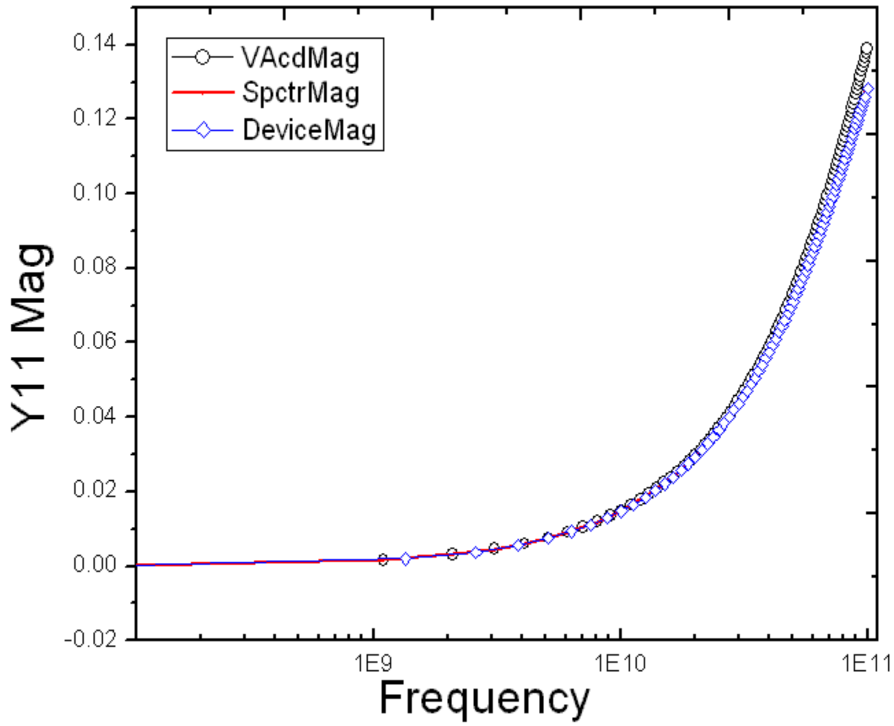


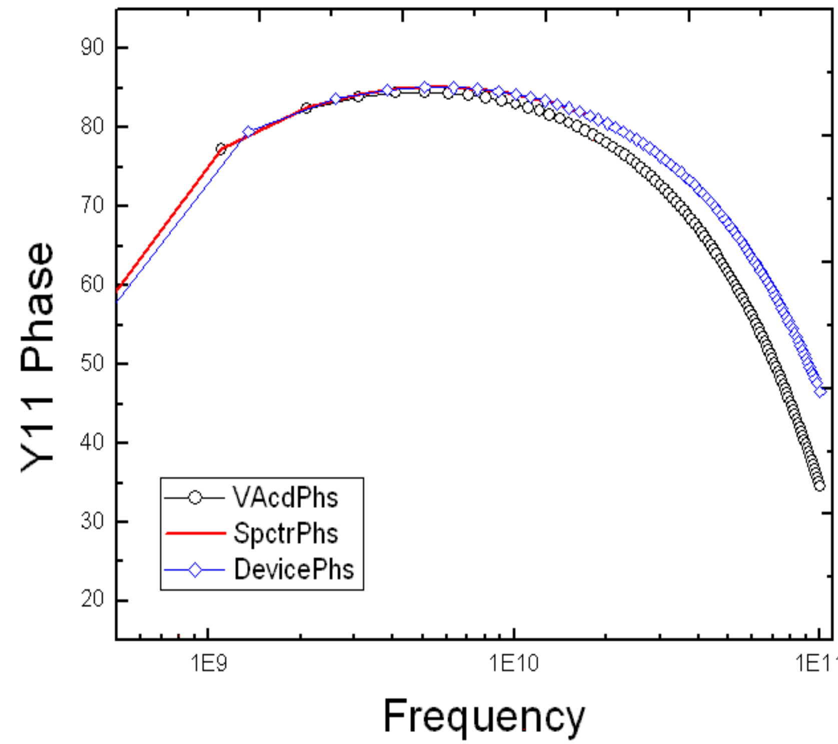
- Weil's approach implementations agree (v2.1 in SPECTRE and DEVICE)
- VA (adj. netw.) deviates from Weil's approach implementation

=> derivative issue?

HICUM comparison: v2.1(Weil) vs. v2.2 (adj. netw.)

Small-signal frequency-domain simulation: Y11





- similar results as for y21
- results with bias **<u>independent</u>** transit and delay times are identical for all implementations

Summary

HICUM/L2 v2.23: will proceed with release

- EDA vendors: use existing simulator-specific NQS code (Weil’s approach)
- joint effort with Agilent:
  define VA language construct for automated generation of "Weil-code"

VA-based NQS effect implementation

- only "LCR"-approach is sufficiently similar to Weil’s approach
  => presently implemented
- first version is available and under test
- can simplify charge-related adjunct network to 1st-order low-pass
- need to clarify small-signal parameter deviation of VA code
- run-time comparison desirable (VA compiled model vs. built-in Weil code)
- further theoretical investigations needed