Mansoura Engineering Journal

Volume 32 | Issue 3 Article 2

12-9-2020

FPGA versus ASIC Implementation of Radix-8 Scalable Montgomery Modular Multiplier.

Atef Ibrahim
Electronics Research Institute., Cairo., Egypt.

Hamed ElSemary
Electronics Research Institute, Cairo, Egypt.

Amen Nassar
Cairo University, Cairo, Egypt

FPGA VERSUS ASIC IMPLEMENTATION OF RADIX-8 SCALABLE MONTGOMERY MODULAR MULTIPLIER

مقارنة تحقيق ضارب مونتجومري المتبقي مختلف الدقة ثماني القاعدة باستخدام مصفوفة البوابات المبرمجة حقليا مع الدوائر المتكاملة المحددة التطبيق

ATEF A. IBRAHIM[1] , *HAMED A. ELSIMARY*[1], *AMEN M. NASSAR*[2]

[1] Electronics Research Institute, Cairo, Egypt,

[2] Cairo University, Cairo, Egypt

الملخص العربي:

إن تحقيق لوغاريتمات التشفير باستخدام تقنية الدوائر المتكاملة محددة التطبيق يكون ذو مرونة أقل منه في حالة التحقيق بواسطة البرمجة. وحيث أن بروتوكولات السرية لا تعتمد على لوغاريتم معين ، لذا أصبح من المرغوب فيه توفير مرونة بدرجة عالية جدا. فأمثل الحلول التي تجمع بين المرونة العالية وكذلك السرعة العالية ، هو تحقيق لوغاريتمات التشفير بواسطة أجهزة قابلة للبرمجة مثل مصفوفة البوابات المبرمجة حقليا. والهدف من هذه الورقة العلمية هو مقارنة – من حيث المساحة والسرعة – تحقيق ضارب مونتجومري المتبقي مختلف الدقة ثماني القاعدة باستخدام مصفوفة البوابات المبرمجة حقليا مع الدوائر المتكاملة المحددة التطبيق وذلك لمعاملات ذات دقة مختلفة.

ABSTRACT

Traditional ASIC implementations have the well known draw-back of reduced flexibility compared to software implementations. Since modern security protocols are increasingly defined to be algorithm independent, a high degree of flexibility with respect to the cryptographic algorithms is desirable. A promising solution which combines high flexibility with the speed and physical security of traditional hardware is the implementation of cryptographic algorithms on reconfigurable devices such as FPGA. In this paper we compare – in terms of area and speed- FPGA implementation of radix-8 scalable Montgomery modular multiplier using retiming technique with ASIC implementation for different word sizes of operands. The simulation data were generated using Mentor Graphics CAD tools.

KEYWORDS: **Montgomery Multiplication, Scalability, FPGA Implementation, ASIC Implementation, Cryptography**

1. INTRODUCTION

Modular multiplication is a widely used operation in cryptography. Several well know applications, such as the decipherment operation of the RSA algorithm[1], the Diffie-Hellman key exchange algorithm[2], as well as some applications currently under development, such as the Digital Signature Standard [3] and elliptic curve

Accepted July 17, 2007

cryptography [4], all use modular multiplication and modular exponentiation. The second operation is often implemented by a series of multiplications and additions [5,6,7,8].

Given the increasing demands on secure communications, cryptographic algorithms will be embedded in almost every application involving exchange of information. Some of these applications, such as smart cards [9] and hand-helds, require hardware restricted in area and power resources [10].

An efficient algorithm to implement modular multiplication is the Montgomery Multiplication algorithm [11]. It has many advantages over ordinary modular multiplication algorithms. The main advantage is that the division step in taking the modulus is replaced by shift operations which are easy to implement in hardware [10].

An aspect of cryptographic applications is that very large numbers are used. The precision varies from 128 and 256 bits for elliptic curve cryptography to 1024 and 2048 bits for applications based on exponentiation [12]. Most of the hardware designs for modular multiplication are fixed precision solutions. That is, the operands can be only of fixed bit-size. Designs that can take operands with an arbitrary precision are researched in the ASIC [13] and the FPGA [8] realms.

A scalable (variable-precision) Montgomery multiplier design methodology was introduced in [13] in order to obtain hardware implementations. This design methodology allows to use a fixed-area modular multiplication circuit for performing multiplication of unlimited precision operands. The design tradeoffs for best performance in a limited chip area were also analyzed in [13]. Extension of this design methodology to higher radices was introduced in [14].

Traditional ASIC implementations, however, have the well known draw-back of reduced flexibility compared to software implementations. Since modern security protocols are increasingly defined to be algorithm independent, a high degree of flexibility with respect to the cryptographic algorithms is desirable. A promising solution which combines high flexibility with the speed and physical security of traditional hardware is the implementation of cryptographic algorithms on reconfigurable devices such as FPGA.

In this paper we compare – in terms of area and speed - FPGA implementation of radix-8 scalable Montgomery modular multiplier using re-timinig technique [14] with ASIC implementation for different word sizes of operands. The simulation data were generated using Mentor Graphics CAD tools.

This contribution is structured as follows. In Section 2 we present the radix-8 Montgomery Modular Multilpication algorithm (R8MM). Section 3 presents the overall organization of the modular multiplier that implements the R8MM. Section 4 shows the simlation results, generated using Mentor Graphics CAD tools. Section 5 concludes the work.

2. R8MM ALGORITHM

The notation used in the presented multiple-word Radix-8 Montgomery Multiplication algorithm (R8MM) is shown below (Fig.1).

Fig. 2 shows the R8MM algorithm, which is an extension of the Multiple-Word High-Radix ($R2^k$) Montgomery Multiplication algorithm ($MWR2^kMM$) presented and proved to be correct in [14].

The Booth recoding [15] was applied to the multiplier X. This recoding scheme translates conventional radix-Λ digits in the set {0, 1, 2, 3 , 4, 5, 6, 7} into the digit set {-4, -3, -2, -1, 0, 1, 2, 3, 4} . The recoded digit Z_j is obtained from the radix-8 multiplier digit $X_j = (x_{3j+2}, x_{3j+1}, x_{3j})$ as:

$$Z_j = \mathrm{Re}coding1(X_j, x_{3j-1}) = -4x_{3j+2} + 2x_{3j+1} + x_{3j} + x_{3j-1}$$

where $j = 0,1,2,....,\left\lceil \frac{N-1}{3} \right\rceil$, N is the multiplier precision.

In order to make the three least-significant bits of the partial product S all zeros, a multiple of the modulus M, namely qM_jM, is added to the partial product. This step is required to make sure that there are no significant bits lost in the right shift operation performed in step 10. To compute the digit qM_j we need to examine the bits from 5 to 3 of the partial

product S generated in step 5 of the R8MM algorithm .

* X – Multiplier , Y- Multiplicand , M – Modulus, S – Partial product
* N - operands precision
* X_j - a single radix-8 digit of X at position j;
* qM_j - quotient digit that determines a multiple of the modulus M to be added to the partial product S;
* w - number of bits in a word of either Y, M or S;
* $e = \left\lceil \frac{N+1}{w} \right\rceil$ - number of words in either Y, M or S;
* NS - number of stages;
* C_a, C_b - carry bits;
* $(Y^{(e-1)},...,Y^{(1)},Y^{(0)})$ - operand Y represented as multiple words;
* $S^{(i)}_{k-1..0}$ - bits k - 1 to 0 of the i^{th} word of S.

Fig.1. Notation

It is shown in [10] that qM_j, as computed in step 6, satisfies the relation $qM_j * M \equiv -S \pmod 8$, which can be rewritten as: $S_{2..0} + qM_j * M_{2..0} \equiv 0 \pmod 8$ and represents the fact that the last 3 bits of S are zeros before the right shift is done in step 10.

The first difficulty in this design comes from the fact that Z and qM can have values that are not powers of 2. As an example, the bit-vector $2Y$ can be produced from Y by left-shifting Y by one bit. However, the bit-vector $3Y$ is produced by adding Y and $2Y$. The latter case requires huge amount of time compared to simple bit-shifting.

For Z the difficult values are 3 and -3 and for qM the difficult values are 3, 5 and 7. One way of implementing the coefficients is to split Z and qM into at least two values each. For example, implementing $Z = 3$, or $3 * Y$, can be done as $(2 * Y) + (1 * Y)$ *or* $(4 * Y) - (1 * Y)$ *without actually* performing the addition or subtraction but using two bit-vectors, $2 * Y$ and $1 * Y$ or $4 * Y$ and $-1 * Y$ in this case. Summing/subtracting the two bit-vectors to obtain the bit-vector for $3Y$ will be an overkill for the computational speed. A better approach is to use two bit-vectors for $((Z_j * Y)^{(*)}$. Same logic applies for qM_j.

Step
1: $S := 0$
$x_{-1} := 0$
$Z_0 := Z1_0 + Z2_0 := Booth(x_{2..-1})$
$qM_0 := (Z1_0 * Y^{(0)}_{2..0} + Z2_0 * Y^{(0)}_{2..0}) * (8 - M^{(0)^{-1}}_{2..0}) \bmod 8$
2: FOR $j := 0$ TO N - 1 STEP 3
3: $Z_{j+3} = Z1_{j+3} + Z2_{j+3} = Booth(x_{j+3+2..j+3-1})$
4: $(C_a, S^{(0)}) := S^{(0)} + Z1_j * Y^{(0)} + Z2_j * Y^{(0)}$
5:
$(C_b, S^{(0)}) := S^{(0)} + q1M_j * M^{(0)} + q2M_j * M^{(0)}$
6:
$qM_{j+3} := q1M_{j+3} + q2M_{j+3} = S^{(0)}_{5..3} * (8 - M^{(0)^{-1}}_{2..0}) \bmod 8$
7: FOR $i := 1$ TO e - 1
8: $(C_a, S^{(i)}) := C_a + S^{(i)} + Z1_j * Y^{(i)} + Z2_j * Y^{(i)}$
9:
$(C_b, S^{(i)}) := C_b + S^{(i)} + q1M_j * M^{(i)} + q2M_j * M^{(i)}$
10: $S^{(i-1)} := (S^{(i)}_{2..0}, S^{(i-1)}_{W-1..3})$
END FOR;
11: $C_a = C_a$ *or* C_b
12: $S^{(e-1)} := signext(C_a, S^{(e-1)}_{W-1..3})$
END FOR;
..... Final result representation

Fig.2. Multiple-word R8MM algorithm.

Some possible combinations for these coefficients are shown in Table 1 where Z_j is represented as $Z1_j$ and $Z2_j$, and qM_j is represented as $q1M_j$ and $q2M_j$.

Table 1
POSSIBLE COMBINATIONS FOR Z_j AND qM_j

Z_j	$Z1_j$	$Z2_j$	qM_j	$q1M_j$	$q2M_j$
-4	-4	0	0	0	0
-3	-4	1	1	1	0
-2	-4	2	2	0	2
-2	-2	0	3	-1	4
-1	-1	0	3	1	2
0	0	0	4	0	4
1	0	1	5	1	4
2	0	2	6	2	4
2	-4	2	7	-1	8
3	-1	4			
4	0	4			

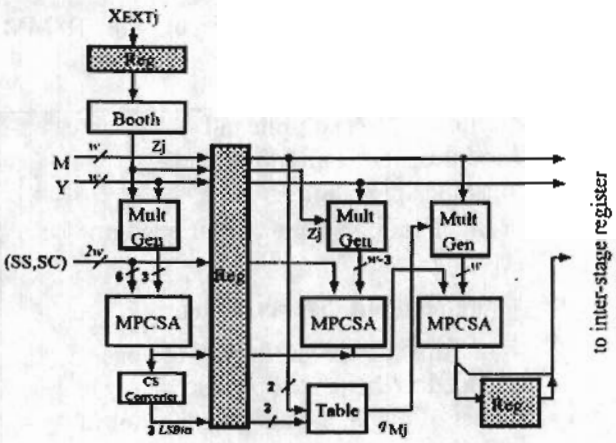

Fig. 3. PE Organization

3. OVERALL ORGANIZATION

The architecture of the modular multiplier that implements the R8MM consists of 3 main blocks; *Datapath (or Kernel)*, *IO & Memory*, and the *Control block*. The computation shown in the R8MM algorithm takes place in the *kernel*[10].

The *kernel* is organized as a pipeline of Processing Elements (PE) [10], separated by registers. Each PE implements one iteration of the R8MM algorithm (steps 3 to 12).

3.1 Radix-8 Processing Element

The radix-8 PE is organized as shown in Fig. 3. The main functional blocks in the PE are: *booth recoding*, *multiple generation (Mult Gen)*, *multi-precision Carry-save adders (MPCSA)*, qM_j *table*, and *registers (shaded boxes)*. The PE operates on *w*-bit words and for this reason the *Mult Gen* and *MPCSA* modules are capable of storing and transferring carry bits from one word to the next. Shifting and word alignment is done by proper combination of signals and registers at the output of the last MPCSA. The design uses a retiming technique explained in [14]. More details about these modules and their operation can be found in [10].

The Processing Element (PE) is divided in two sections. The first section (before the register) computes only the three least-significant (LS) bits of each word of $S + Z1_j Y + Z2_j Y$. One can observe that qM_j depends on three LS bits of the data coming from the preceding PE in the pipeline: $(S_{2..0}^{(0)})$ and $Y^{(0)}$, and the recoded digit Z_j. The word size for S needs to be at least 6 bits in order to have the three LS bits of S generated as early as possible for the next PE.

A stage consists of a PE and a register. At each clock cycle, one word of Y, M, SS, and SC is applied as inputs to a stage. The multiplier digits X_i are transferred to PEs at specific times. The newly computed words of SS and SC, together with words of Y and M, are propagated by each stage to the next stage. This way, small PEs work concurrently to perform several iterations of the R8MM algorithm.

4. SIMULATION RESULTS

The simulation data were generated using Mentor Graphics CAD tools. The radix-8 design presented in this paper was described in VHDL and simulated in ModelSim for functional correctness. A simulation results of this algorithm are shown in Fig.4 for the input operands $X = A = 3$, $Y = B = 4$ and $M = 5$, which results in $SS = result = 2$, $SC = carry = 1$ (for N = 8, NS = 3, w = 4). Note

that, these input operands and result are in Montgomery domain.

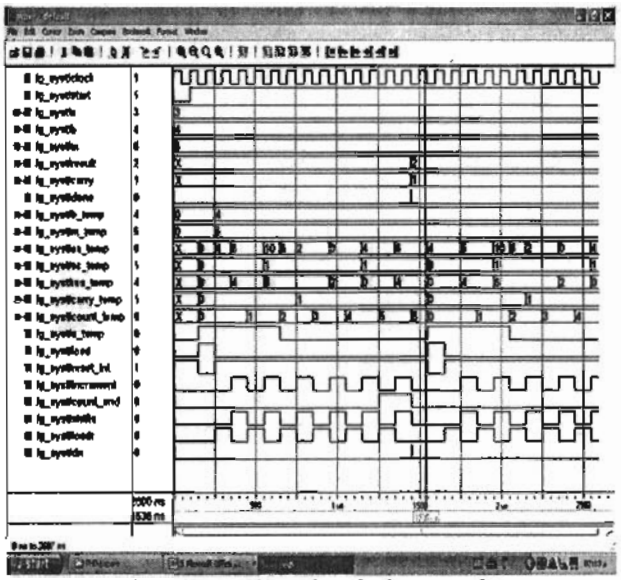

Fig.4. Function simulation results

4.1 ASIC Implementation

Radix-8 design was synthesized using Leonardo synthesis tool for *AMI*05- *slow* (0.5 *μm CMOS* technology with hierarchy preserved) provided in the ASIC Design Kit (ADK) from the same company. It has to be noted that the ADK has been developed for educational purposes. however, it provides a consistent environment for comparison between the designs, and a reasonable approximation of the system performance.

4.1.1 Area Estimation for Radix-8 Kernel

The area of the kernel depends on the two design parameters: number of stages in the pipeline (*NS*), and the word size (*w*) of the operands (Y , M) and the result (S). the total area of the kernel is given by [10]:

$$A\ker nel_{R8} = 92 * NS * w + 269 * NS - 9.42 * w - 35.5 \quad (1)$$

Table 2 is constructed using Eq. 1. The area estimates are given in terms of 2-input NOR gate.

4.1.2 Time Estimation for Radix-8 Kernel

The total computational time for the kernel is a product of the number of clock cycles it takes and the clock period. Table 3 shows the critical path delay - measured in *ns* - as a function of the number of stages in the pipeline (*NS*), as well as the word size (*w*) of the operands. As can be seen from the Table, the critical path delay in some cases remains constant even if the number of stages is increased. This attributed to using carry-save logic.

A word of Y , M, and S propagates through the pipeline for $(2 * NS + 1)$ clock cycles. The speed of scanning the bits of X for radix-8 is three bits per stage. Based on these observations, Eq. 2 represents the total number of clock cycles needed for R8MM [10].

$$Tclks = \begin{cases} \left\lceil \frac{N}{3 * NS} \right\rceil * (2 * NS + 1) + \left\lceil \frac{N}{w} \right\rceil + 1 , & if \left\lceil \frac{N}{w} \right\rceil \le 2 * NS \\ \left\lceil \frac{N}{3 * NS} \right\rceil * \left(\left\lceil \frac{N}{w} \right\rceil + 1 \right) + 2 * NS , & if \left\lceil \frac{N}{w} \right\rceil > 2 * NS \end{cases} \quad (2)$$

The total computational time is obtained by multiplying *Tclks* by the corresponding critical path delay (clock period) shown in Table 3, which was obtained from synthesis tools.

Table 2
AREA IN NUMBER OF NOR GATES FOR RADIX-8 KERNEL (ASIC)

NS	Word Size (w)				
	8	16	32	64	128
1	894	1630	3102	6046	11934
2	1899	3371	6315	12203	23979
3	2904	5112	9528	18360	36024
4	3909	6853	12741	24517	48069
5	4914	8594	15954	30674	60114
7	6924	12076	22380	42988	84204
8	7929	13817	25593	49145	96249
9	8934	15558	28806	55302	108294
12	11949	20781	38445	73773	

13	12954	22522	41658	79930
15	14964	26004	48084	92244
16	15969	27745	51297	98401
20	19989	34709	64149	
26	26019	45155	83427	

Table 3
CRITICAL PATH DELLAY FOR RADIX-8 KERNEL (ASIC)

NS	Word Size (w)				
	8	16	32	64	128
1	10.7	10.3	13.1	18.9	20.2
2	10.8	12.1	14.4	20.5	30.4
3	10.9	12.5	15.7	23.0	
4	11.0	12.9	17.0	25.4	
5	11.1	12.7	17.6		
6	11.1	13.5	18.2		
7	11.2	14.3	18.7		
8	11.2	14.9	19.2		
9	11.2	15.1			
10	11.2	15.2			
11	11.2	15.3			
12	11.2	15.4			
13	11.3	15.4			
14	11.3	15.4			
15	11.3	15.5			
20	11.4				
26	13.0				

4.2 FPGA IMPLEMENTATION

Radix-8 design was synthesized using Leonardo synthesis tool for Xilinx Virtex –II technology.

4.2.1 Area Results

The area in FPGA is given in terms of Configurable Logic Blocks (CLBs). Each CLB approximately has 172 2-input NOR gate. Table 4 shows the area – in number of 2-input NOR gates - as a function of the number of stages in the pipeline (*NS*), as well as the word size (*w*) of the operands.

4. 2.2 Time Results

Table 5 shows the critical path delay (measured in *ns*) as a function of the number of stages in the pipeline (*NS*), as well as the word size (*w*) of the operands. As can be seen from the Table, the critical path delay in some cases remains constant even if the number of stages is increased. This attributed to using carry-save logic.

Table 4
AREA IN NUMBER OF NOR GATES FOR RADIX-8 KERNEL (FPGA)

NS	Word Size (w)				
	8	16	32	64	128
1	1204	2236	3440	7396	11980
2	2064	3612	7224	14620	24908
3	2924	5332	10664	20812	37324
4	3612	7396	13588	26144	53320
5	4472	8944	17028	32164	
7	6192	12212	22390	45752	
8	6708	13588	25628	51428	
9	7740	15308	29068	55556	
12	8944	18920	35260		
13	9632	20468	38700		
15	11696	25284	48676		
16	10320	27004	52460		
20	11180	33540			
26	15136	40764			

Table 5
CRITICAL PATH DELAY FOR RADIX-8 KERNEL (FPGA)

NS	Word Size (w)				
	8	16	32	64	128
1	10.42	10.61	11.35	12.35	11.75
2	10.52	10.62	11.38	12.36	11.80
3	10.62	11.13	11.38	12.36	11.80
4	10.65	11.13	11.48	12.36	11.81
5	10.62	11.14	11.48	12.38	11.83
6	10.63	11.34	11.48	12.38	11.84
7	10.63	11.35	11.49	12.38	11.85
8	10.64	11.35	11.49	12.39	11.85
9	10.64	11.36	11.51	12.39	11.87
10	10.10	11.44	11.52	12.39	11.87
11	10.20	11.44	11.52		
12	10.21	11.54	11.53		
13	11.20	11.54	11.53		
14	11.22	11.56	11.55		
15	11.22	11.56	11.55		
20	11.32	11.61	11.56		
26	11.32	11.61	11.56		

5. CONCLUSION

R8MM was implemented on ASIC technology (*AMI*05 - *slow)* as well as FPGA (Xilinx Virtex – II) technology. The Montgomery multiplier implemented is a variable-precision solution. FPGA is selected since it can be easily reconfigured for different word size. Thus, their design area increases correspondingly with the word size used. As can be seen from Tables 2, and 4, there is a significant increase in chip area of ASIC when w is less than 16 bit and NS greater than 2, and there is a significant increase in chip area of FPGA when w is greater than 16 bit for all NS . the timing results suggest that the proposed ASIC implementation can perform as well as the FPGA implementation. Whereas the ASIC implementation cannot be reconfigured, this proposed word size solution design allows the system to work on any precision so long as the precision does not exceed certain limit.

REFERENCES

[1] L. Adleman and A. Shamir, "A method for obtaining digital signature and public-key cryptosystems," *Comm. ACM*, vol. 21, no. 2, pp. 120-126, February 1978.

[2] M..E. Hellman , "New directions on cryptography," *IEEE transactions on Information Theory*, vol. 22, pp. 644-654, November 1976.

[3] National Institute for Standards and Technology, "Digital signature standard (dss)," Tech. Rep., FIPS PUB 186-2, January 2000.

[4] N. Koblitz, "Elliptic curve cryptosystems.," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, January 1987.

[5] A.J. Menezes, *Applications on finite fields.*, Kluwer Academic Publishers, Boston, MA, 1993.

[6] B.S. Kaliski, Ç.K. Koç, and T. Acar, "Analysing and comparing Montgomery multiplication algorithms," *IEEE Micro*, vol. 16, no. 3, pp. 26-33, June 1996.

[7] T. Hamano, "O(n)-depth circuit algorithm for modular exponentiation," in *IEEE 12th Symposium on Computer Arithmetic*. 1995, pp. 188-192, IEEE Computer Society Press, Los Alamitos, CA.

[8] C. Paar and T. Blum, "Montgomery modular exponentiation on reconfigurable hardware," in IEEE 14th Symposium on Computer Arithmetic. 1999, pp. 70-77, IEEE Computer Society Press, Los Alamitos, CA.

[9] D. M. Raihi and D. Naccache, "Cryptographic smart cards," IEEE Micro, vol. 16, no. 3, pp. 14-23, June 1996.

[10] G. Todorov, "ASIC design, implementation and analysis of a scalable high-radix Montgomery multiplier," Master thesis, Oregon State University, USA, December 2000.

[11] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of Computation*, vol. 44, no. 170, pp. 519-521, April 1985.

[12] Ç.K. Koç, E. Savas, and A. F. Tenca, "A scalable and unified multiplier architecture for finite fields GF(p) and FG(2^n)," in *Cryptographic Hardware and Embedded Systems*. 2000, Lecture Notes in Computer Science, Springer, Berlin, Germany.

[13] Ç.K. Koç and A. F. Tenca, "A word-based algorithm and architecture for montgomery multiplication," in *Cryptographic Hardware and Embedded Systems*, C. Paar and Ç. Koç , Ed. 1999, number 1717 in Lecture Notes in Computer Science, pp. 94-108, Springer, Berlin, Germany.

[14] A. F. Tenca, G. Todorov, and Ç.K. Koç, "High-radix design of a scalable modular multiplier," in *Cryptographic Hardware and Embedded Systems* - CHES 2001, Ç.K. Koç and C. Paar, Eds. 2001, Lecture Notes in Computer Science, No. 1717, pp. 189-206, Springer, Berlin, Germany.

[15] A. D. Booth, "A signed binary multiplication technique," *Q. J. Mech. Appl. Math.*, vol. 4, no. 2, pp. 236-240, 1951.

LOAD CHARACTERISTICS EFFECT ON DYNAMIC VOLTAGE STABILITY ANALYSIS IN HVDC SYSTEMS

تأثير خصائص الحمل على تحليل إستقرار الجهد الديناميكى فى نظم الجهد الفائق المستمر

I. Bedir[1] A. A. Lotfy[2] G.E.M. Aly[1]

1 Tanta University, faculty of Engineering, Tanta, Egypt.
2 Arab Academy for Science &Technology & Maritime Transport, Alexandria, Egypt.

ملخص. تقدم هذه الورقه البحثيه طريقه جديده لتحليل تأثير نموذج الحمل على إستقرار الجهد فى نظم الجهد الفائق المستمر عند طرفى الموحد و العاكس. و قد تم أخذ التغيرات الديناميكيه لنظم التيار المستمر و المتحكمات فى الإعتبار عند مختلف ظروف التشغيل. و قد تم تطبيق و إستخدام نموذج الإشاره الصغيره للإتزان للتعرف على حدود إتزان النظام فى حالة وجود الحمل الإستاتيكى والديناميكى . كما تم حساب متغيرات نظم التيار المستمر و المتردد عند النسب المختلفة للقصر الفعال لبيان تأثر إستقرار النظام. و قد تم التحقق من النتائج بإستخدام المحاكاه الغير خطيه.

Abstract: The modeling of loads has a significant effect on the accuracy of dynamic voltage stability analysis of HVDC system. This paper investigates the dynamic nature of voltage instability considering static and dynamic load models. The load effect at different control modes of HVDC system is considered for different configurations of single infeed HVDC systems at different effective short circuit ratios. The results are validated using nonlinear simulations.

Keywords: Load Characteristics, HVDC, SIF, SIFAC, Bifurcations.

I. INTRODUCTION

The concept of voltage instability have been observed in AC systems when operating close to its steady state stability limit, also the voltage stability is related to special load locations. Converter terminals used for HVDC system can be seen as a special load which may cause voltage instability [1].

Different configurations of HVDC systems are used today at different places around the world [2]. The main configurations of HVDC systems are single-infeed (SIF) [3,5,6], single-infeed with a parallel AC line (SIFAC) [3,7] and multi-infeed systems [7,9].

Several researchers tackled the voltage instability problem for SIF [3-6], other researches developed these techniques to be suitable for SIFAC [3,7]. A static analysis of SIF considering static load effect was given in [3,4]. In the dynamic analysis given in [6] a simple representation of DC line and a simple RL circuit were considered for only two control modes. In [5] a model suitable for SIF systems sensitivity analysis was given, nevertheless, power flow effect and load power at converter bus were not considered through stability study.

In this paper, a detailed model of SIF and SIFAC systems incorporating static and dynamic load models is introduced. Nonlinear simulation is used to validate model results.

SYMBOLS:

$CESCR_{Inv}$, $CESCR_{Rect}$	Critical effective short circuit ratio for both rectifier and inverter, respectively.
CC,Cβ, CD A, CP, CDV	Constant current, constant beta, constant delay angle, constant power and constant DC voltage controllers, respectively.
Δx_{c1} and Δx_{c2}	Output of integral branch of PI Controller for both rectifier and inverter, respectively
α and β	Rectifier firing and inverter advance angles, respectively
δ_1, δ_2	Rectifier and inverter bus phase angles.
P_{d1}, P_{d2}, Q_{d1}, Q_{d2}	DC line active power transfers and reactive power consumed at rectifier and inverter side, respectively.
P_{Sr}, P_{Si}, Q_{Sr}, Q_{Si},	Active and reactive static load power at rectifier and inverter, respectively.
n_{pv}, n_{qv}	Active/reactive voltage dependent power order.
K_{pc}, k_{qc}	The active and reactive power load constants.
K_{pv}, k_{qv}	Voltage dependent active and reactive load constants.
K_{pi}, k_{qi}	The current active and reactive load constants.
K_{pz}, k_{qz}	Impedance active and reactive load constants.
P_{Lr}, P_{Li}, Q_{Lr}, Q_{Li}	Active and reactive dynamic load power at both rectifier and inverter, respectively.
FL, CL, ZL, VL	Fixed, constant current, impedance, and voltage dependent loads, respectively.

Accepted September 11, 2007

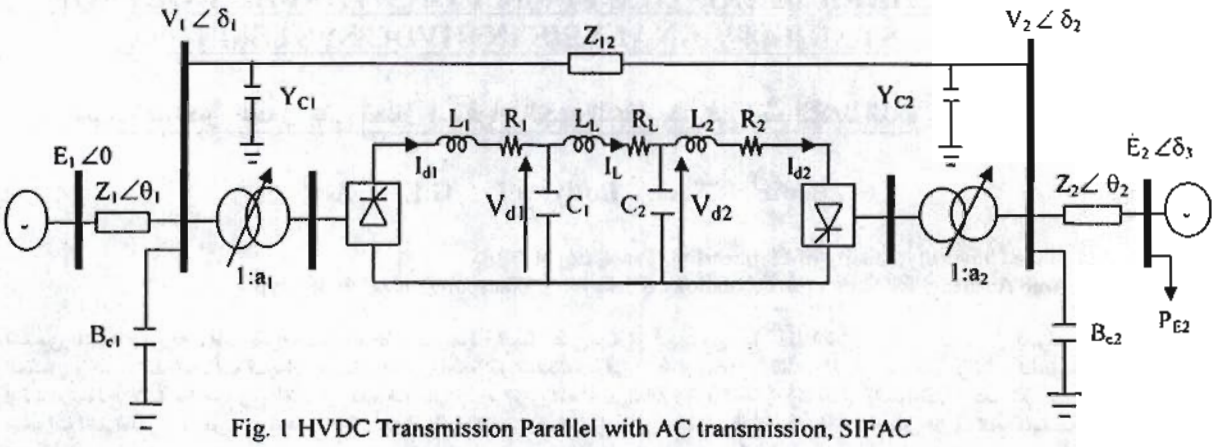

Fig. 1 HVDC Transmission Parallel with AC transmission, SIFAC

II. SYSTEM MODEL

The HVDC system for both SIF and SIFAC consists of the following parts:

1. DC System Model

The DC network includes converters (rectifier and inverter), smoothing reactors and DC transmission line. The DC transmission line is represented by its π–equivalent. The DC network is shown in the middle of Fig. 1. The DC system and DC controller differential equations are similar to that presented in [4, 8].

2. AC System Model

The active and reactive power flow through AC lines in both rectifier and inverter for SIF can be written as follows:

$$P_{acj} = V_j^2 Y_j \cos\theta_j + V_j E_j Y_{jE} \cos(\theta_{jE} - \delta_j)$$
$$Q_{acj} = -V_j^2 Y_j \sin\theta_j - V_j E_j Y_{jE} \sin(\theta_{jE} - \delta_j) \quad (1)$$

Where; $\bar{Y}_j = 1/\bar{Z}_j + jB_{Cj} = Y_j \angle \theta_j$ and

$\bar{Y}_{jE} = 1/\bar{Z}_j = Y_{jE} \angle \theta_{jE}$,

$\bar{Z}_j = R_j + jX_j$

j =1 for rectifier and 2 for inverter.

3. AC-DC Power Flow equations

a) No load at converter buses

The system algebraic equations for SIF can be written as follows:

$$P_{ac1} + P_{d1} = 0$$
$$P_{ac2} - P_{d2} = 0 \quad (2)$$
$$Q_{ac1} + Q_{d1} = 0$$
$$Q_{ac2} + Q_{d2} = 0 \quad (3)$$

Equations (2) and (3) can be replaced by equations (4) and (5) to be suitable for SIFAC:

$$P_{ac1} + P_{d1} + P_{12} = 0$$
$$P_{ac2} - P_{d2} + P_{21} = 0 \quad (4)$$
$$P_{ac3} + P_{E2} = 0$$

$$Q_{ac1} + Q_{d1} + Q_{12} = 0$$
$$Q_{ac2} + Q_{d2} + Q_{21} = 0 \quad (5)$$

where;

$$P_{12} = V_1 V_2 Y_{12} \cos(\theta_{12} + \delta_2 - \delta_1) = -P_{21}$$
$$P_{ac3} = E_2^2 Y_3 \cos\theta_3 + E_2 V_2 Y_{2E} \cos(\theta_{2E} + \delta_2 - \delta_3)$$
$$Q_{12} = -V_1 V_2 Y_{12} \sin(\theta_{12} + \delta_2 - \delta_1) = -Q_{21}$$
$$\bar{Y}_{12} = -\frac{1}{\bar{Z}_{12}} = Y_{12} \angle \theta_{12} = Y_{21} \angle \theta_{21}$$
$$\bar{Y}_3 = \frac{1}{\bar{Z}_2} = Y_3 \angle \theta_3 \quad (6)$$

b) Static Load at Converter Buses

In this work the general static load model presented in [3] was used to modify the stability analysis model presented in [8]. The active and reactive static load at converter bus can be taken as follows:

$$P_{S_{r,j}} = k_{pc} + k_{pi} V_j + k_{pz} V_j^2 + k_{pv} V_j^{n_{pv}}$$
$$Q_{S_{r,j}} = k_{qc} + k_{qi} V_j + k_{qz} V_j^2 + k_{qv} V_j^{n_{qv}} \quad (7)$$

where j = 1 for rectifier and 2 for inverter.

The active and reactive static load power at rectifier and inverter are to be added to Equations (2) and (3) for SIF or to Equations (4) and (5) for SIFAC

c) Dynamic Load

The dynamic active and reactive loads $P_{L_{r,i}}, Q_{L_{r,i}}$ are given by [6]

$$P_{L_{r,i}} = \frac{1}{T_{p_{r,i}}}[x_{p_{r,i}} + \frac{1}{2}k_{pL}V_i^2]$$
$$Q_{L_{r,i}} = \frac{1}{T_{q_{r,i}}}[x_{q_{r,i}} + \frac{1}{2}k_{qI}V_i^2] \quad (8)$$

The active and reactive load functions are respectively given by:

$$\dot{x}_{p_{r,i}} = P_{S_{r,i}} - P_{L_{r,i}}$$
$$\dot{x}_{q_{r,i}} = Q_{S_{r,i}} - Q_{L_{r,i}} \quad (9)$$

The active and reactive static and dynamic load at rectifier and inverter are to be added to Equations (2) and (3) for SIF or to Equations (4) and (5) for SIFAC

4. Small Signal Stability Model

The system differential equations can be linearized to obtain the state space model as.

$$\dot{x}_{DC} = A\, x_{DC} + B\, u_{DC} \quad (10)$$

where:

$$x_{DC} = [\Delta V_{r1}, \Delta V_{r2}, \Delta I_{d1}, \Delta I_{d2}, \Delta V'_{d1}, \Delta V'_{d2}, \Delta I_L]$$
$$u_{DC} = \begin{cases} [\Delta\delta_1, \Delta\delta_2, \Delta I'_1, \Delta I'_2] \text{ for SIF} \\ [\Delta\delta_1, \Delta\delta_2, \Delta\delta_3, \Delta I'_1, \Delta I'_2] \text{ for SIFAC} \end{cases}$$

By linearizing equations (2) and (3) for SIF or (4) and (5) for SIFAC, the state space form of algebraic equations is obtained as.

$$0 = Cx_{DC} + Du_{DC} \quad (11)$$

Where, "C" and "D" are the Jacobian submatrices. Assuming that D remains nonsingular along system trajectories as the system parameters vary, then equations (10) and (11) are reduced to [3,5].

$$\dot{x}_{DC} = A' x_{DC} \quad (12)$$

Where $A' = A - BD^{-1}C$.

Equation (12) represents the small signal stability model of DAE suitable for SIF and SIFAC system. Voltage stability analysis is carried out by computing Eigenvalues of the system state matrix A'.

Most of the voltage instability problems are related to bifurcation. These bifurcations characterized by changes of Eigenvalues of the system equilibria as certain parameters change in the system. The main types of bifurcation are saddle node bifurcation (SN) which occurs when one Eigenvalue become zero and Hobf bifurcation (HP) which occurs when a pair of complex Eigenvalues cross the imaginary axis [3-4, 11-13] The effect of voltage instability is greater at the AC bus connected to the converter operated at low short circuit ratio.

III. CASE STUDY

The data of the HVDC system used to implement the proposed technique is given in Table 1.

Table 1 AC and DC systems data (p.u.)

AC system Data (p.u.)

Bus No	E	R_c	L	R + j X	B_c
1	1.1	0.115	0.0052	0 + j0.2857	0.4
2	1.1	0.115	0.0052	0 + j0.3333	0.6
Z_{12} = 0.2 + j 0.6 p.u., $y_{c1} = y_{c2}$ = 0.01 p.u.					

DC Line Data (p.u.)

R_l	L_L	C_1	C_2
0.04462	0.000823	0.00027	0.000272

The HVDC system strength can be measured by the system effective short circuit ratio which is a parameter used to study system instability [6]:

$$ESCR_i = \frac{1}{Z_i} - B_{ci} \quad (13)$$

CASE (a) Static Load

The static load data at rectifier and inverter are illustrated in Table 2.

Table 2 Static load data in p.u.

k_{pv}	k_{pi}	k_{pz}	k_{pv}	n_{pv}
0.1	0.02	0.02	0.02	1.5
k_{qc}	k_{qi}	k_{qz}	K_{qv}	n_{qv}
0.1	0.01	0.01	0.01	1.5

Fig. 2 illustrates the p-v nose curve due to change of active load power at inverter bus of SIF. The AC line voltage at both rectifier and inverter decrease with an increase of static active load power up to a maximum power of 2.522 p.u ..

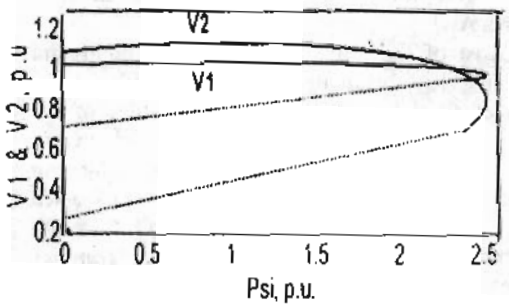

Fig. 2 The p-v nose curve at rectifier and inverter due to change of active load power at inverter (SIF adopting CC/Cβ)

Fig. 3 illustrates the p-v nose curve due to change of active load power at rectifier bus of SIFAC. The AC line voltage at both rectifier and inverter decrease with an increase of static active load power up to a maximum power of 1.031 p.u..

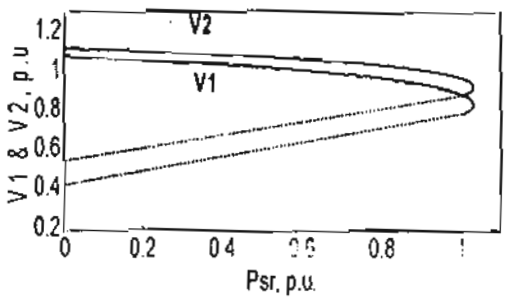

Fig. 3 The p-v nose curve at rectifier and inverter due to change of active load power at rectifier (SIFAC adopting CDA/CC).

Figs. 4 and 5 illustrate the P_{dr} against V_r and P_{dr} against I_{dr} curves at rectifier side for different types of static active and reactive loads respectively. The maximum value of DC power transfer varies according to the applied static load types. The maximum DC power transfer at different static loads is shown in Tables 3 and 4 for SIF and SIFAC, respectively.

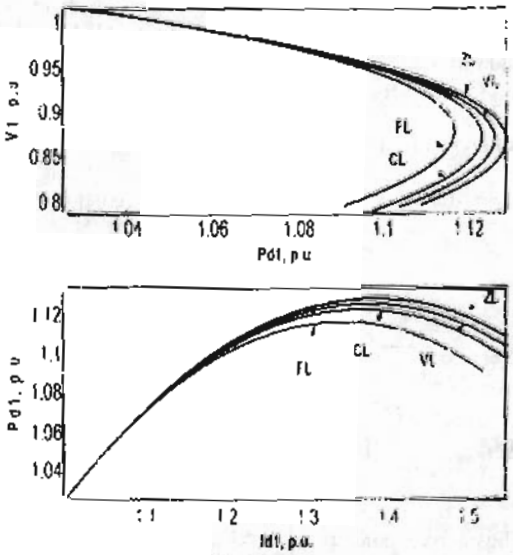

Fig. 4 The P_{dr}-V_r and P_{dr}-I_{dr} curves at rectifier at different types of static active load power (SIF adopting CDA/CC).

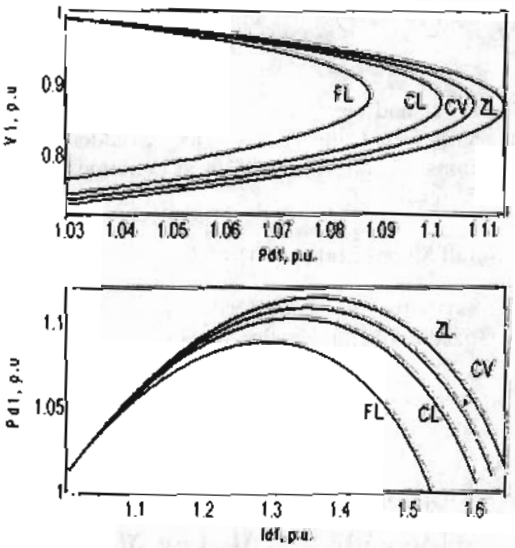

Fig. 5 The P_{dr}-V_r and P_{dr}-I_{dr} curves at rectifier at different types of static active load power (SIFAC adopting CDA/CC).

Table 3 Maximum DC Power Transfer for SIF

Load Type	Max. P_{dr} at Static Active Power Loads	Max. P_{dr} at Static Reactive Power Loads
FL	1.117	1.035
CL	1.123	1.056
ZL	1.129	1.073
VL	1.126	1.065

Table 4 Maximum DC Power Transfer for SIFAC

Load Type	Max. P_{dI} at Static Active Power Loads	Max. P_{dI} at Static Reactive Power Loads
FL	1.160	1.088
CL	1.164	1.102
ZL	1.168	1.114
VL	1.166	1.108

Fig. 6 illustrates the behavior of AC line voltage at rectifier bus versus the voltage dependent active and reactive load coefficients, at different AC line voltage dependent power orders, respectively. The system strength increases with increasing of voltage order, which must be greater than 1 [10].

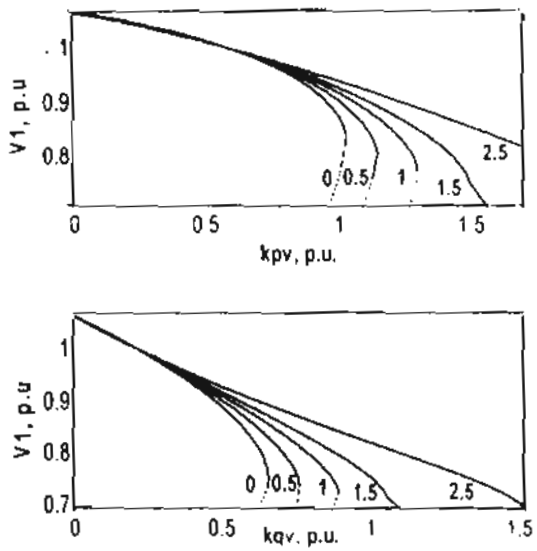

Fig. (6) AC line voltage versus the voltage dependent load coefficients for different power orders.

Reaching a critical effective short circuit ratio may be due to Saddle node (SN) or Hopf (HP) bifurcations, or power flow failure (PF). In case of load at rectifier bus, the power order coefficient of voltage dependent portion of static load positively affects the stability at this bus due to the associated reduction of this load portion. As shown in Fig 7, the effect of n_{qv} change is relatively more noticeable compared with that of n_{pv} due to the direct bearing of the reactive power on stability

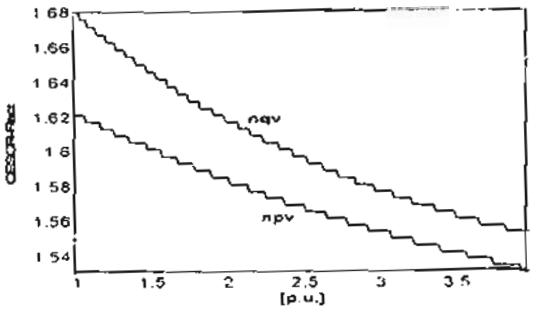

Fig. 7 $CESCR_{Rect}$ versus n_{pv1} and n_{qv1} for CDA/CC control mode (K_{pv1}, K_{qv1}=0.1)

Fig. 8 illustrates the effect of increasing k_{pc}, k_{qc}, k_{pi}, k_{qi}, k_{pz}, k_{qz}, k_{pv} and k_{qv} on $CESCR_{Rect}$. It illustrates that the $CESCR_{Rec}$ increases with an increase in either of the load constants. Nevertheless, the increase of k_{qc} and k_{pc} yielded the most significant bearing on system instability.

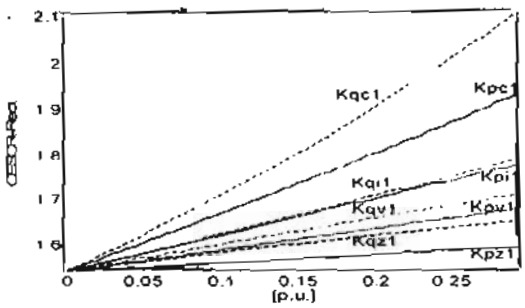

Fig. 8 $CESCR_{Rect}$ against k_{pc1}, k_{qc1}, k_{pi1}, k_{qi1}, k_{pz1}, k_{qz1}, k_{pv1} and k_{qv1} for CDA/CC

Fig. 9 shows the expected deterioration of stability at the inverter bus due to its reactive loadings. A similar loading at the rectifier bus positively affects the inverter's bus stability due to the reduction of dc power transferred and the associated reduction of reactive power needed for the commutation process. The DC line's performance is thus reduced on behalf of stability.

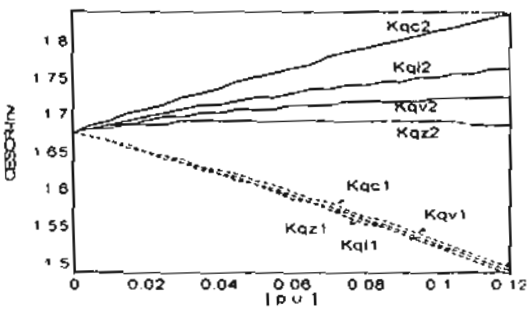

Fig. 9 $CESCR_{Inv}$ versus $k_{qc1,2}$, $k_{qi1,2}$, $k_{qv1,2}$, and $k_{qz1,2}$ for CC/Cβ control mode Load at Rectifier and Inverter bus, respectively.

Fig. 10 corresponds to an SIF stable case at SCR_{Rect} of 2.7776. The system adopts CDV/CC control mode with an increase of 0.01 p.u. in the current order of the inverter's CC controller. The rectifier current oscillates around a stable node. The responses of AC line voltages at both rectifier and inverter buses cause the shown subsequent changes in static active and reactive load power

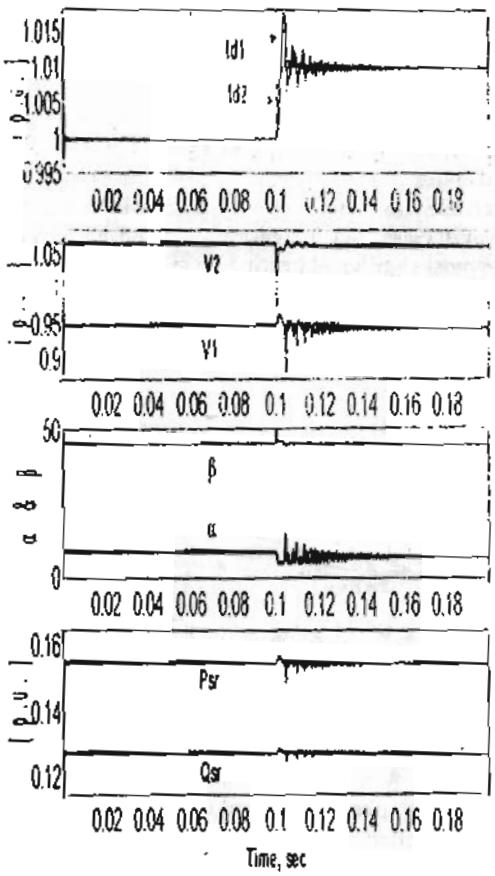

Fig (10) Time response of system variables due 0.01 increase in inverter current order (SIF adopting CDV/CC control at $ESCR_{Rect}$ 2.7776).

Fig. (11) corresponds to an unstable SIF configuration. The system adopts CC/Cβ control mode at SCR_{Inv} of 1.66091 (HP). Fig. 11.a shows the time response of AC line voltages, rectifier firing angle and static load powers at inverter bus. The variables are found to oscillate around an unstable node, which is evident from the phase plane of AC and DC line voltages against DC line current at inverter bus as shown in Fig. 11.b.

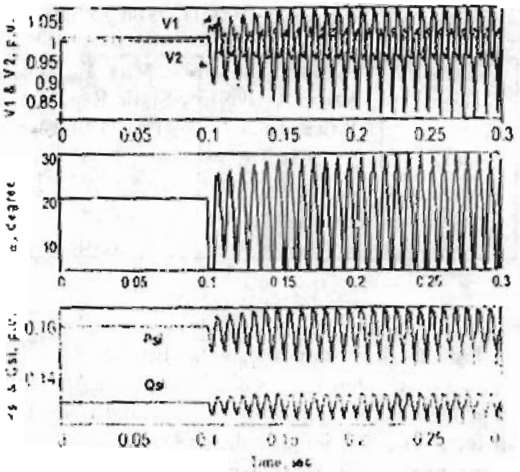

Fig 11 a Time responses

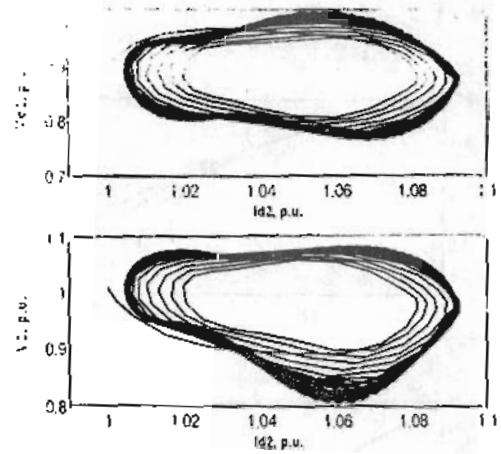

Fig 11 b Phase plane of V_{d2}, I_{d2} and V_2, I_{d2}

Fig 11 Time responses and phase planes due to 0.05 increase in rectifier current order (SIF adopting CC/Cβ at $ESCR_{Inv}$ of 1.66091)

To compare between the stability performances of SIF and SIFAC under load conditions, the responses of the later are studied at $ESCR_{Inv}$ value below that rendered the SIF configuration unstable. Fig 12 shows the phase plane of AC and DC line voltages against DC line current at inverter bus for an SIFAC configuration at $ESCR_{Inv}$ of 1.6609 which illustrates that the variables oscillate around a stable node. The system remained stable due to the active and reactive power transfer capability from rectifier bus to inverter bus through the parallel AC line, which raises the voltage at the inverter bus.

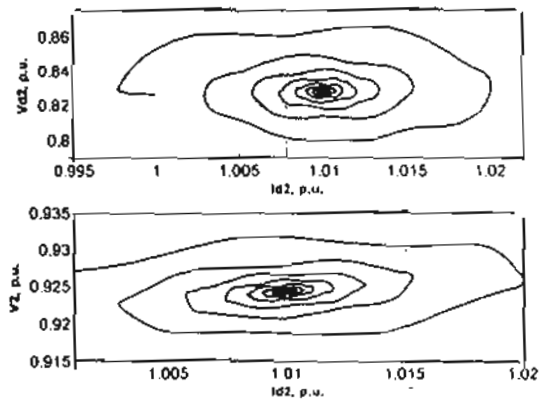

Fig. 12 Phase planes of V_{d2}, I_{d2} and V_2, I_{d2} due to 0.05 increase in rectifier current order (SIFAC adopting CC/Cβ at $ESCR_{Inv}$ of 1.6609)

The different values of critical ESCR at different control modes for both SIF and SIFAC are presented in Table 5.

Table 5 CESCR with Static Load at rectifier or inverter bus.

Control Mode	SIF	SIFAC
CDA/CC	1.89516 (SN)	1.8799 (SN)
CC/Cβ	1.66091 (HP)	1.3081 (PF)
CDV/ CC	2.6544 (PF)	1.3314 (PF)
CP/Cβ	1.7693 (HP)	1.2512 (SN)
CDV/CP	2.8248 (PF)	2.2717 (PF)

CASE (b) Dynamic and Static Load

The dynamic active and reactive load and the static load are applied at the converter bus. The load data is shown in Table 6.

Table 6 Dynamic & Static load data in p.u.

k_{pc}	n_{pv}	k_{pL}	T_p
0.05	2	0.01	0.04
k_{qc}	n_{qv}	k_{qL}	T_q
0.05	2	0.01	0.04

Fig. 13 corresponds to an unstable case with SIF configuration at SCR_{Rect} of 1.06954 (Hopf bifurcation). The system adopts CP/Cβ control mode. Fig. 13.a shows the time responses of DC line current at both rectifier and inverter sides due to a change of power order by 0.001 p.u which oscillate around an unstable node. It shows the time response of AC line voltage at inverter bus. It shows also the time response of active and reactive dynamic and static load power. The active and reactive dynamic load power is largely affected by system change rather than that of static load. Fig. 13.b shows the phase plane of AC and DC line voltages against DC line current at inverter bus which illustrates that they oscillate around an unstable node point.

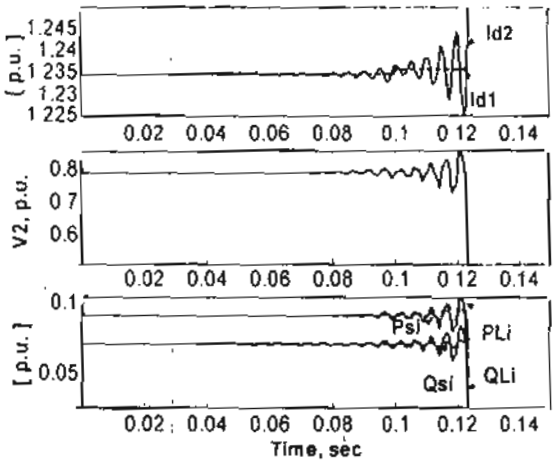

Fig 13.a Time responses due to rectifier power order increase of 0.001 (SIF adopts CP/Cβ at $ESCR_{Inv}$ 1.06954)

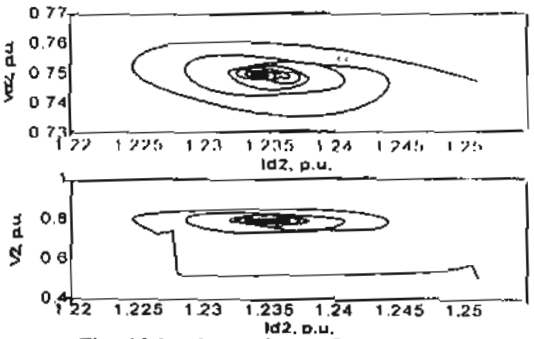

Fig. 13.b phase plane of V_{d2}, I_{d2} and V_2, I_{d2} due to rectifier power order increase of 0.001 (SIF adopts CP/Cβ at $ESCR_{Inv}$ 1.06954)

Fig. 14 corresponds to a stable case with SIFAC configurations at SCR_{Rect} of 1.7032. The system adopts CP/Cβ control mode. It shows the time responses due to a larger change of power order of 0.05 p.u.

The system variables oscillate around a stable node in spite of the larger perturbation due to the compensating effect of the AC line.

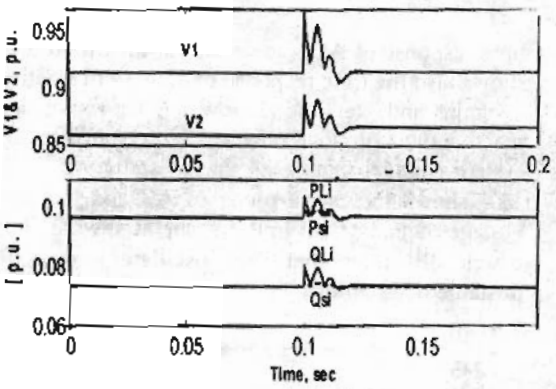

Fig 14 system variables' response due to increase of rectifier power order by 0.05 (SIFAC adopts CP/Cβ , $ESCR_{Inv}$ = 1.7032)

The system's critical ESCR values at different control mode of SIF and SIFAC with dynamic and static load are presented in Table 7.

Table 7 CESCR with Static and Dynamic Loads

Control Mode	SIF	SIFAC
CDA/CC	1.8336 (SN)	1.9955 (PF)
CC/Cβ	1.5085 (HP)	1.4421 (HP+SN)
CDV/ CC	2.5351 (PF)	1.7391 (PF)
CP/Cβ	1.06954 (HP)	1.3194 (SN)
CDV/CP	2.2199 (PF)	2.8936 (PF)

IV. CONCLUSION

The effect of different types of static and dynamic loads at both rectifier and inverter terminals on the stability of the system were presented. The analysis has been carried out for different HVDC system configurations and the maximum DC power transfer at different loading conditions have been assessed. The results verified the expected negative bearing of reactive loading at inverter on stability as well as the positive effect of the AC line in SIFAC configuration. Furthermore, the analysis revealed certain operating conditions where system's stability was seemingly enhanced at the cost of de-rated system's performance; as the case of rectifier side loading. Further work is recommended to propose indices that adequately consider HVDC System's stability and performance as well.

REFERENCES

[1] Pilotto L. A. S., Szechtman M., Hammad A. E. "Transient AC Voltage Related Phenomena for HVDC Schemes Connected to Weak AC Systems", *IEEE Transactions on Power Delivery. Vol. 7, No. 3, PP. 1396-1404, July (1992)*

[2] Vijay K. Sood, "HVDC AND FACTS Controllers Applications of Static Converters in Power Systems", *Kluwer Academic Publishers, (2004).*

[3] Aik, D. L. H. and Anderson G., "Influence of Load Characteristics on The Power/Voltage Stability of HVDC Systems, Part 1: Basic Equations and Relationships", *IEEE Transactions on Power Delivery. Vol. 13, No. 4, PP.1437-1444, Oct. (1998).*

[4] Aik, D. L. H. and Anderson G., "Influence of Load Characteristics on The Power/Voltage Stability of HVDC Systems, Part 2: Stability Margin Sensitivity", *IEEE Transactions on Power Delivery. Vol. 13, No. 4, PP. 1445-1452, Oct. (1998).*

[5] Padiyar K. R. and Rao S. S., "Dynamic Analysis of Voltage in AC-DC Systems", *Elsevier Science Ltd. Electrical Power & Energy Systems, vol. 18, No. 1, PP. 11-18, (1996)*

[6] Aik, D. L. H. and Anderson G., "Nonlinear Dynamics In Hvdc Systems", *IEEE Transactions on Power Delivery. Vol. 14, No. 4, PP.1417-1426, October (1999).*

[7] Aik, D. L. H. and Anderson G., "Voltage Stability Analysis Multi-Infeed HVDC Systems", *IEEE Transactions on Power Delivery. Vol. 12, No. 3, PP. 1309, July (1997).*

[8] Lotfy A. A., Bedir I., Abed-El-Kareem M. E. and Aly G. E. M., "Dynamic Voltage Stability Analysis in HVDC Systems", Alexandria Eng. Journal, Vol. 46, No.2, *PP* 111-119, March (2007).

[9] Zhao C., and Sun Y., "Study on Control Strategies to Improve The Stability of Multi-Infeed HVDC Systems Applying VSC-HVDC", *IEEE CCECE/CCGEL, Ottawa, PP. 2253-2257 May (2005).*

[10] Pai M. A., Sauer P. W., Lesieutre B. C., and Adapa R., "Structural Stability Power System – Effect of Load Models", *IEEE Trans. on Power systems; Vol. 10, No. 2, PP. 609-615, May (1995).*

[11] Mithulananthan N., Ganizares C. A. and Reeve J., "Indices to Detect Hopf Bifurcations in Power Systems", *NAPS-2000, PP. 1-7, (2000)*

[12] Harb A. M. and Abed-Jabar N., "Controlling Hopf Bifurcations and Chaos in A Small Power Systems", *Elsevier Science Ltd. Chaos, Solutions and Fractals, 18, PP. 1055-1063, (2003)*

[13] Hranilovic S. and Canizares C. A., "Transcritical and Hopf Bifurcations in AC/DC Systems", *Prog. Bulk Power Voltage Phenomena, Davos, Switzerland, PP. 105-114, August (1994).*