รายการอ้างอิง

ภาษาไทย

กิติพจน์ สิทธิเลิศพิศาล , วงจรกรองแอกทีฟแบบอนุกรมสำหรับลดกระแสฮาร์มอนิกและรักษาระดับแรงดัน , การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18 , 2538

โสภณ สมัยรัฐ , ระบบควบคุมมอเตอร์เหนี่ยวนำแบบเวกเตอร์ด้วยไมโครคอนโทรลเลอร์ ,วิทยานิพนธ์มหาบัณฑิตจุฬาลงกรณ์มหาวิทยาลัย , 2538

ภาษาอังกฤษ

F.Z.Peng ,"A New Approach to Harmonic Compensation in Power System", Dissertation Nagaoka University of Technology , February 1990

_____H.Akagi , A.Nabae , "A Study of Active Filters Using Quad-Series Voltage-Source PWM Converters for Harmonic Compensation",IEEE/PESC'87 RECORD, 1987, pp. 204-242

_____H.Akagi , A.Nabae , "Analysis and Suppression for Harmonic Instability in Phase-Controlled Thyristor Converters" , Paper of Technical Meeting on Semiconductor Power Converter, IEE , Japan , SPC-90-5, pp. 39-49

_____H.Akagi , A.Nabae , "Compensation Characteristics of the Combined System of Shunt Passive and Series Active Filter" , Paper of Technical Meeting on Semiconductor Power Converter, IEE , Japan , SPC-89, pp. 959-967

_____H.Akagi , A.Nabae , "Novel Harmonic Power Filter",IEEE/PESC'88 RECORD, 1988, pp. 1151-1159

IEEE Standard 519-1992 , "IEEE Recommened Practices And Requirements for Harmonic Control in Electrical Power Systems", IEEE, 1992, pp. 78

Shoji FUKUDA and Takayoshi ENDOH, "Control Method for A Combined Active Filter System Employing a Current Source Converter and A High Pass Filter" , PCC-Yokohama , 1993 , pp. 621-628

S.Y. Choe K.Heumann, "Current Control Strategies for Active Power Filter with Inductive Energy Storage",PCC-Yokohama, 1993, pp. 817-824

Toshifumi ISE , Zhaoan WANG , Yoshishige MURAKAMI , "Compensation Characteristics of The Active Filter using A Current Source PWM Converter",PCC-Yokohama, 1993, pp.791-798

Tokuo OHNISHI and Haruyuki YAMAUCHI , "A Novel Control Strategy of Active Filter", IPEC-Tokyo'90, 1990,pp. 783-790

ภาคผนวก ก

ภาคผนวก ก

การหาฟังก์ชันโอนย้ายบนแกนนิ่ง *Ga(s)* จากฟังก์ชันโอนย้ายบนแกนหมุน *A(s)*

ในการพิจารณาฟังก์ชันโอนย้ายของวงจรกรองเพื่อคำนวณหาองค์ประกอบฮาร์มอนิกของกระแสนั้น เมื่อเราพิจารณาบนแกนหมุนจะได้ว่า

$$\begin{bmatrix} i_{sd}^{*} \\ i_{sq}^{*} \end{bmatrix} = A(s) \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \qquad \text{(ก.1)}$$

สมมุติให้ฟังก์ชันโอนย้าย A(s) =M(s)/N(s) โดยที่ M(s) , N(s) เป็นพหุนาม (polynomials) อันดับที่ m และ n ตามลำดับ เราจะสามารถเขียนสมการ(ก.1) ได้เป็น

$$N(s) \cdot \begin{bmatrix} i_{sd}^{*} \\ i_{sq}^{*} \end{bmatrix} = M(s) \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \qquad \text{(ก.2)}$$

จากสมการ(2.4) เราสามารถหาตวามสัมพันธ์ระหว่างการหาอนุพันธ์บนแกนหมุนกับการหาอนุพันธ์บนแกนนิ่งได้ดังต่อไปนี้คือ

$$s\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = s\left[\begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \right] \qquad \text{(ก.3)}$$

โดยที่

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_{su} \\ i_{sv} \\ i_{sw} \end{bmatrix}$$

จากสมการ(ก.3) จะได้ว่า

$$s\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = s\begin{bmatrix} i_{s\alpha} \cdot \cos(\omega t) + i_{s\beta} \cdot \sin(\omega t) \\ -i_{s\alpha} \cdot \sin(\omega t) + i_{s\beta} \cdot \cos(\omega t) \end{bmatrix}$$

$$=\begin{bmatrix} -i_{s\alpha}\cdot\omega\cdot\sin(\omega t)+\cos(\omega t)\cdot si_{s\alpha}+i_{s\beta}\cdot\omega\cdot\cos(\omega t)+\sin(\omega t)\cdot si_{s\beta} \\ -i_{s\alpha}\cdot\omega\cdot\cos(\omega t)-\sin(\omega t)\cdot si_{s\alpha}-i_{s\beta}\cdot\omega\cdot\sin(\omega t)+\cos(\omega t)\cdot si_{s\beta} \end{bmatrix}$$
$$=\begin{bmatrix} \cos(\omega t)\cdot(si_{s\alpha}+i_{s\beta}\cdot\omega)+\sin(\omega t)\cdot(si_{s\beta}-i_{s\alpha}\cdot\omega) \\ -\sin(\omega t)\cdot(si_{s\alpha}+i_{s\beta}\cdot\omega)+\cos(\omega t)\cdot(si_{s\beta}-i_{s\alpha}\cdot\omega) \end{bmatrix}$$
$$=\begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.4)$$

ในทำนองเดียวกันเราสามารถคำนวณได้ว่า

$$s^n\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}=\begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}^n\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.5)$$

ดังนั้นจากสมการ(ก.2) เนื่องจากทั้ง *M(s)* และ *N(s)* เป็นพหุนาม เราจึงสามารถใช้ความสัมพันธ์(ก.5) แทนในแต่ละพจน์ของ *M(s)* และ *N(s)* ได้ และเราจะสามารถเขียนใหม่ได้เป็น

$$\begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}\cdot N(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\cdot\begin{bmatrix} i^*_{s\alpha} \\ i^*_{s\beta} \end{bmatrix}=$$
$$\begin{bmatrix} \cos(\omega t) & -\sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix}\cdot M(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\cdot\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.6)$$

เมื่อพิจารณาฟังก์ชันโอนย้ายในการหากระแสส่วนฮาร์มอนิกบนแกนนิ่ง จะได้ว่า

$$N(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\begin{bmatrix} i^*_{s\alpha} \\ i^*_{s\beta} \end{bmatrix}=M(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.7)$$

ดังนั้นเราจะได้ฟังก์ชันโอนย้ายบนแกนนิ่งเป็น

$$\begin{bmatrix} i^*_{s\alpha} \\ i^*_{s\beta} \end{bmatrix}=\left[N(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\right]^{-1}*\left[M(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}}\right]\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.8)$$

เมื่อย้ายไปพิจารณาบนแกน positive, negative sequence โดยมีนิยามการแปลงแกนดังนี้คือ

$$\begin{bmatrix} i^*_{sp} \\ i^*_{sn} \end{bmatrix}=[C]\begin{bmatrix} i^*_{s\alpha} \\ i^*_{s\beta} \end{bmatrix},\ \begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix}=[C]\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \qquad (ก.9)$$

โดยที่

$$[C]=\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}$$

เราจะสามารถเขียนฟังก์ชันโอนย้ายตามสมการ(ก.8) บนแกน positive, negative sequence ได้เป็น

$$\begin{bmatrix} i_{sp}^* \\ i_{sn}^* \end{bmatrix} = [C]\left[N(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}} \right]^{-1} [C]^{-1}[C]\left[M(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}} \right][C]^{-1}\begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix}$$

$$= \left\{ [C]\left[N(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}} \right][C]^{-1} \right\}^{-1} \left\{ [C]\left[M(\lambda)\Big|_{\lambda=\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}} \right][C]^{-1} \right\}\begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix}$$

$$= \left[N(P)\Big|_{P=[C]\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}[C]^{-1}} \right]^{-1}\left[M(P)\Big|_{P=[C]\begin{bmatrix} s & \omega \\ -\omega & s \end{bmatrix}[C]^{-1}} \right]\begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix}$$

$$= \left[N(P)\Big|_{P=\begin{bmatrix} s-j\omega & 0 \\ 0 & s+j\omega \end{bmatrix}} \right]^{-1}\left[M(P)\Big|_{P=\begin{bmatrix} s-j\omega & 0 \\ 0 & s+j\omega \end{bmatrix}} \right]\begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix}$$

$$\begin{bmatrix} i_{sp}^* \\ i_{sn}^* \end{bmatrix} = \begin{bmatrix} A(s-j\omega) & 0 \\ 0 & A(s+j\omega) \end{bmatrix}\begin{bmatrix} i_{sp} \\ i_{sn} \end{bmatrix} \qquad \text{(ก.10)}$$

สมการ(ก.10) ถึงแม้ว่าจะดูเหมือนมีลักษณะเป็นสมการอิสระ 2 สมการ แต่ในความเป็นจริงแล้วเมื่อให้ s = jw จะเห็นได้ว่าสมการทั้งสองเป็นเพียงค่าสังยุค (conjugate) ของกันและกัน เราจึงเลือกใช้เพียงสมการเดียวก็เพียงพอ และเนื่องจากระบบแกนอ้างอิงที่เราใช้นั้นเป็นแบบ positive sequence เราจึงใช้ความสัมพันธ์ที่ว่า

$$\begin{aligned} \left[i_{s\alpha}^* + j\, i_{s\beta}^* \right] &= A(s-j\omega)\left[i_{s\alpha} + j\, i_{s\beta} \right] \\ \left[i_{s\alpha}^* + j\, i_{s\beta}^* \right] &= Ga(s)\left[i_{s\alpha} + j\, i_{s\beta} \right] \end{aligned} \qquad \text{(ก.11)}$$

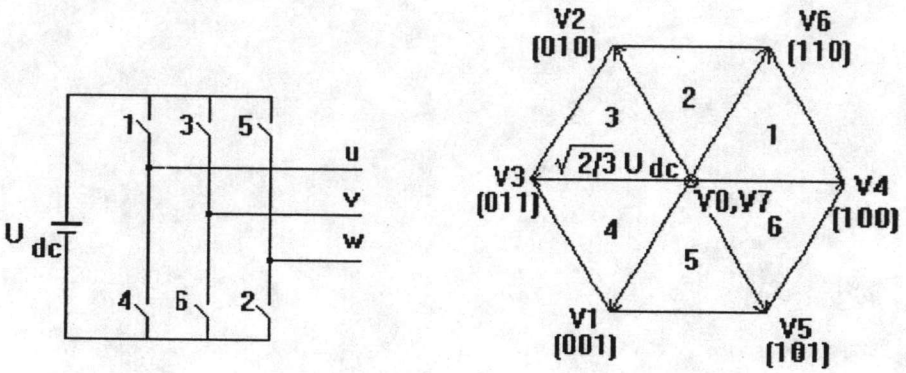
หรือกล่าวได้อีกนัยหนึ่งคือ *Ga(s)* แสดงถึงความสัมพันธ์ระหว่างเวกเตอร์กระแสแหล่งจ่าย $\vec{i}_s$ และเวกเตอร์กระแสฮาร์มอนิก $\tilde{i}_s$ เมื่อมองเวกเตอร์เป็นจำนวนเชิงซ้อนนั้นเอง

ภาคผนวก ข

ภาคผนวก ข

อินเวอร์เตอร์แบบสเปซเวกเตอร์ของแรงดัน (โสภณ สมัยรัฐ, 2538)

โครงสร้างอินเวอร์เตอร์ที่ใช้ในการควบคุมแบบปรับความกว้างพัลส์ (PWM) จะประกอบด้วยแหล่งจ่ายไฟฟ้ากระแสตรง Udc และ สวิตช์กำลัง 6 ตัว ซึ่งสามารถจะสร้างเวกเตอร์แรงดันได้ 8 แบบ (V0-V7) ตามรูปที่ ข.1



(ก) (ข)

รูปที่ ข.1 (ก) แบบจำลองชุดอินเวอร์เตอร์ (ข) เวกเตอร์แรงดันของชุดอินเวอร์เตอร์

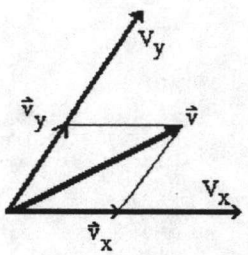
จากการที่แรงดันที่สามารถสร้างได้มีเพียง 8 แบบ เราจึงไม่สามารถที่จะสร้างเวกเตอร์แรงดันตามต้องการได้โดยตรง แต่จะใช้วิธีการสร้างเวกเตอร์แรงดันที่มีค่าเฉลี่ยเท่ากับแรงดันที่ต้องการแทน ดังนั้นจะต้องทำการพิจารณาว่าเวกเตอร์แรงดันที่ต้องการสร้างไปตกอยู่ในเซกเตอร์ใดใน 6 เซกเตอร์ตามที่แสดงในรูปที่ ข.1 เมื่อทราบว่าเวกเตอร์แรงดันที่ต้องการตกอยู่ในเซกเตอร์ใดแล้วก็จะทำการแตกเวกเตอร์แรงดันนั้นไปอยู่ในแนวของเวกเตอร์องค์ประกอบข้างเคียงของเซกเตอร์นั้นๆตามแสดงในรูปที่ ข.2 เช่นถ้าเวกเตอร์ที่ต้องการไปตกอยู่ในเซกเตอร์ที่ 1 ก็จะทำการแตกเวกเตอร์แรงดันที่ต้องการไปอยู่ในแนงของเวกเตอร์ V4 และ V6 ดังนั้นในกรณีทั่วไปเราจะแตกเวกเตอร์แรงดันที่ต้องการลงในแนวของเวกเตอร์ที่ใช้ในการสวิตช์ ได้แรงดันเป็น $\vec{v} = \vec{v}_x + \vec{v}_y$ โดยที่ $\vec{v}_x$ เป็นเวกเตอร์องค์ประกอบในแนว Vx และ $\vec{v}_y$ เป็นเวกเตอร์องค์ประกอบในแนว Vy

ถ้าเรากำหนดให้เวลาที่ใช้ในการสวิตช์ทั้งหมดเท่ากับ T_s , t_x เท่ากับเวลาในการต่อวงจรของสวิตช์ตามเวกเตอร์ V_x และ t_y เท่ากับเวลาในการต่อวงจรของสวิตช์ตามเวกเตอร์ V_y ส่วนช่วงเวลาที่เหลือจะเป็นของเวกเตอร์ศูนย์ (V_0, V_7) แผนผังเวลาในการสวิตช์แสดงได้ในรูปที่ ข.3 ดังนั้นเราจะได้ค่าเฉลี่ยของเวกเตอร์แรงดัน $\bar{v}$ ในช่วงเวลา T_s เป็น

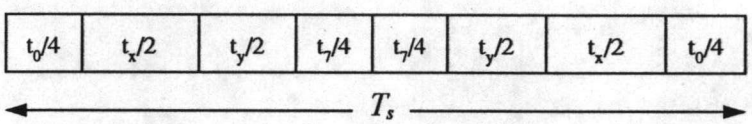
$$\vec{v} \cdot T_s = V_x \cdot t_x + V_y \cdot t_y + V_0 \cdot \frac{t_0}{2} + V_7 \cdot \frac{t_0}{2}$$

$$T_s = t_x + t_y + t_0 \qquad \text{(ข.1)}$$

$$\therefore \quad \vec{v} = \left(\frac{t_x}{T_s}\right) V_x + \left(\frac{t_y}{T_s}\right) V_y$$



รูปที่ ข.2 ส่วนประกอบของเวกเตอร์แรงดัน



รูปที่ ข.3 แผนผังเวลารูปแบบการสวิตช์

จากความสัมพันธ์ข้างต้น เราสามารถคำนวณเวลาในการสวิตช์ (t_x ,t_y และ t_0) ได้ว่า

$$t_x = \frac{|\vec{v}_x|}{V_x} \cdot T_S \qquad , \qquad t_y = \frac{|\vec{v}_y|}{V_y} \cdot T_S$$

$$t_0 = T_S - t_x - t_y \qquad \text{(ข.2)}$$

โดยที่ T_s = 1 คาบเวลาที่ใช้ในการสวิตช์

t_x = เวลาในการต่อวงจรของสวิตช์ตามเวกเตอร์ V_x

t_y = เวลาในการต่อวงจรของสวิตช์ตามเวกเตอร์ V_y

t_0 = เวลาในการต่อวงจรของสวิตช์ตามเวกเตอร์ V_0 , V_7

ซึ่งจะทำให้ได้เวกเตอร์แรงดันที่มีค่าเฉลี่ยตามที่ต้องการเท่ากับ

$$\vec{v} = \left(\frac{t_x}{T_s}\right) V_x + \left(\frac{t_y}{T_s}\right) V_y = \vec{v}_x + \vec{v}_y \qquad \text{(ข.3)}$$

<u>การชดเชยค่าแรงดันในการสร้างสัญญาณ PWM กรณีระดับแรงดันบัสไฟตรงมีค่าเปลี่ยนแปลง</u>

เวกเตอร์แรงดันของอินเวอร์เตอร์ V1~V6 ที่ใช้ในการสร้างสัญญาณ PWM นั้นจะมีขนาดขึ้นอยู่กับขนาดของแรงดันบัสไฟตรงซึ่งสามารถคำนวณได้เป็น

$$\left|V_{1\sim6}\right| = \sqrt{\tfrac{2}{3}}U_{dc} \qquad \text{(ข.4)}$$

ดังนั้นในกรณีที่ค่าแรงดันที่บัสไฟตรงมีค่าเปลี่ยนแปลงไป เราจึงจำเป็นที่จะต้องปรับเปลี่ยนค่าขนาดของเวกเตอร์แรงดัน V1~V6 ตามไปด้วยจึงเป็นการสะดวกและดีกว่าที่จะเขียนสมการ(ข.2) ใหม่เป็น

$$t_x = \sqrt{\frac{3}{2}} \cdot \frac{\left|\vec{v}_x\right|}{U_{dc}} \cdot T_S \quad , \quad t_y = \sqrt{\frac{3}{2}} \cdot \frac{\left|\vec{v}_y\right|}{U_{dc}} \cdot T_S$$

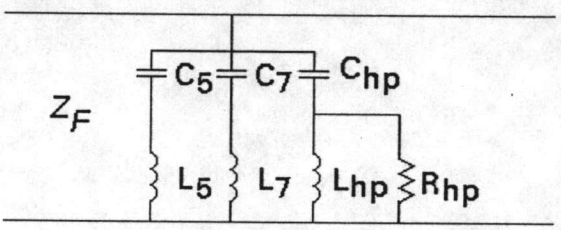
$$t_0 = T_S - t_x - t_y \qquad \text{(ข.5)}$$

ภาคผนวก ค

ภาคผนวก ค

การออกแบบวงจรกรองกำลังแบบพาสซีฟ

การนำเอาวงจรกรองกำลังแบบพาสซีฟมาใช้งานจำเป็นที่จะต้องคำนึงถึงลักษณะของโหลดที่ต่ออยู่ ในงานวิจัยนี้ได้พิจารณาโหลดที่มีลักษณะเป็นวงจรควบคุมเรียงกระแส 3 เฟส ซึ่งโหลดดังกล่าวจะมีฮาร์มอนิกในลำดับเลขคี่ลำดับที่ 5,7,11 เป็นต้น ดังนั้นวงจรกรองกำลังแบบพาสซีฟที่จะนำมาใช้งานจะประกอบไปด้วยวงจรกรองแบบช่องบาก (notch filter) สำหรับฮาร์มอนิกในลำดับที่ 5, 7 และ วงจรกรองผ่านความถี่สูง ตามแสดงดังวงจรสมมูลหนึ่งเฟสในรูปที่ ค.1



รูปที่ ค.1 วงจรสมมูลหนึ่งเฟสของวงจรกรองกำลังแบบพาสซีฟ

วงจรกรองกำลังแบบพาสซีฟนอกจากจะทำหน้าที่ดักจับฮาร์มอนิกแล้วยังทำหน้าที่สร้างกำลังงานรีแอกทีฟ (reactive power) ให้กับระบบไฟฟ้า ดังนั้นการออกแบบวงจรกรองกำลังแบบพาสซีฟจะคำนึงถึงค่ากำลังงานรีแอกทีฟด้วย ในงานวิจัยนี้ได้ออกแบบโดยคิดโหลดวงจรควบคุมเรียงกระแส 3 เฟสขนาด 2 kW ซึ่งมีค่ากำลังงานรีแอกทีฟ 1.54 kVAR แรงดันขาเข้าระหว่างสาย 208 V ซึ่งเมื่อพิจารณาจากสมการกำลังงานรีแอกทีฟของตัวเก็บประจุจะได้ว่า

$$C_{require} = \frac{Q}{V^2 * \omega} = \frac{1.54 \times 10^3}{208^2 \times 2 \times \pi \times 50} \qquad \text{(ค.1)}$$

โดยที่ $C_{require}$ แทนค่าตัวเก็บประจุที่จำเป็นในการชดเชยกำลังงานรีแอกทีฟคิดเป็นค่าต่อเฟส

Q แทนค่ากำลังงานรีแอกทีฟที่โหลดสร้าง

V คือแรงดันระหว่างสาย

จากสมการ(ค.1)จะได้ว่า

$$C_{require} = 113 \quad \mu F$$

ซึ่งเราจะทำการกระจายค่าตัวเก็บประจุนี้ไปตามวงจรกรองฮาร์มอนิกอันดับ 5,7 และวงจรกรองผ่านความถี่สูง กล่าวคือ

$$C_5 + C_7 + C_{hp} = C_{require} = 113 \quad \mu F \qquad \text{(ค.2)}$$

ในส่วนคุณสมบัติการกรองของวงจรนั้นเราจะพิจารณาเริ่มต้นจากค่าอิมพีแดนซ์ที่ความถี่สูงของวงจรกรองผ่านความถี่สูงในวงจรกรองกำลังพาสซีฟ โดยคิดจากการแบ่งการไหลของกระแสฮาร์มอนิกที่แบ่งไหลระหว่างแหล่งจ่ายและวงจรกรองกำลังพาสซีฟ ในขณะที่วงจรกรองกำลังแอกทีฟสร้างแรงดันเสมือนความต้านทาน 65 โอห์ม โดยจะออกแบบให้กระแสฮาร์มอนิกที่แบ่งไหลไปยังแหล่งจ่ายมีค่าไม่เกิน 20 % ซึ่งจากความสัมพันธ์ที่ว่า

$$i_{sh} \cdot K = i_{fh} \cdot \left| Z_f \right| \qquad \text{(ค.3)}$$

และจากการประมาณว่าที่ความถี่สูงค่าอิมพีแดนซ์ของวงจรกรองกำลังพาสซีฟจะมีค่าประมาณ R_{hp} ทำให้เราเขียนสมการ(ค.3)ได้ใหม่เป็น

$$i_{sh} \cdot K \approx i_{fh} \cdot R_{hp} \qquad \text{(ค.4)}$$

จากการแทนค่า K = 65 ทำให้เราสามารถคำนวณค่า R_{hp} ได้มีค่าเป็น 16.25 โอห์ม ซึ่งเมื่อคิดกลับไปอยู่ทางด้านทุติยภูมิที่แรงดัน 208 โวลต์ จะได้ค่า R_{hp} เป็น

$$R_{hp} \approx \quad 4.8 \quad \Omega \qquad \text{(ค.5)}$$

ในส่วนการออกแบบวงจรกรองผ่านความถี่สูงจะอาศัยความสัมพันธ์ตามสมการ(ค.6)และ(ค.7)

$$\omega_o = \frac{1}{R_{hp} C_{hp}} \qquad \text{(ค.6)}$$

$$m = \frac{L_{hp}}{R_{hp}^2 C_{hp}} \qquad \text{(ค.7)}$$

โดย ω_0 คือความถี่หักมุมของวงจรกรองผ่านความถี่สูง

m คือตัวประกอบการหน่วงของวงจรกรองผ่านความถี่สูง

การออกแบบเลือกใช้ $m = 0.7$, $\omega_o = 2 \cdot \pi \cdot 500 \quad rad/s$ ซึ่งจะเป็นค่าที่ทำให้ค่าอิมพีแดนซ์ของวงจรกรองผ่านความถี่สูงนี้มีค่าต่ำในช่วงความถี่ฮาร์มอนิกอันดับที่ 13 เมื่อแทนค่าลงในสมการ (ค.6) และ สมการ(ค.7) แล้วทำการแก้สมการหาค่าตัวแปรได้คือ

$$L_{hp} = 0.675 \text{ mH} \quad , \quad C_{hp} = 75 \ \mu\text{F} \qquad \text{(ค.6)}$$

ในส่วนวงจรกรองแบบช่องบาก เราต้องเลือกค่าตัวเหนี่ยวนำและตัวเก็บประจุให้เกิดเรโซแนนซ์ที่ความถี่ฮาร์มอนิกอันดับที่ 5 และ 7 ตามลำดับ ดังนั้นเราจึงได้ว่า

$$L_5 * C_5 = 1/\omega_5 \quad , \quad \omega_5 = 2*\pi*50*5$$

$$L_7 * C_7 = 1/\omega_7 \quad , \quad \omega_7 = 2*\pi*50*7 \qquad \text{(ค.7)}$$

เพื่อความง่ายเราจะเลือกให้ค่า L_5 , L_7 มีความสัมพันธ์ดังนี้คือ

$$L_5 \approx L_7 \qquad \text{(ค.8)}$$

อาศัยสมการ(ค.2),(ค.7) และสมการ (ค.8) เราสามารถแก้สมการหาค่าตัวแปรได้ดังนี้

$$L_5 = 16.211 \text{ mH} \quad , \quad C_5 = 25 \ \mu\text{F}$$

$$L_7 = 16.906 \text{ mH} \quad , \quad C_7 = 13 \ \mu\text{F}$$

$$L_{hp} = 0.675 \text{ mH} \quad , \quad C_{hp} = 75 \ \mu\text{F}$$

$$R_{hp} = 4.8 \ \Omega$$

เมื่อนำค่าที่ได้จากการออกแบบมาสร้างเป็นวงจรกรองกำลังพาสซีฟจริง และนำวงจรกรองกำลังพาสซีฟที่ได้สร้างขึ้นไปทดสอบกับโหลดวงจรควบคุมเรียงกระแส 3 เฟส ทำการวัดค่าแรงดันที่ตกคร่อมวงจรกรองกำลังพาสซีฟ และวัดค่ากระแสที่ไหลลงไปยังวงจรกรองกำลังพาสซีฟ จากนั้นทำการคำนวณกลับเพื่อหาขนาดของอิมพีแดนซ์ของวงจรกรองกำลังพาสซีฟ ซึ่งได้ค่าตามที่แสดงดังในตารางที่ ค.1

ตารางที่ ค.1 การคำนวณหาค่าอิมพีแดนซ์ของวงจรกรองกำลังพาสซีฟ

ความถี่ฮาร์มอนิกอันดับที่	แรงดันคร่อมวงจรกรองกำลังพาสซีฟ [V]	กระแสที่ไหลเข้าวงจรกรองกำลังพาสซีฟ [A]	อิมพีแดนซ์ของวงจรกรองกำลังพาสซีฟ [Ω] (ด้านแรงดัน 208 V)	อิมพีแดนซ์ของวงจรกรองกำลังพาสซีฟ [Ω] (ด้านแรงดัน 380 V)
1	118	4.14	28.5	95.13
5	0.956	0.1268	7.54	25.16
7	0.868	0.0782	11.1	37.05
11	0.57	0.21	2.71	9.06
13	0.218	0.0846	2.58	8.6
17	0.256	0.0978	2.617	8.74

ภาคผนวก ง

ภาคผนวก ง

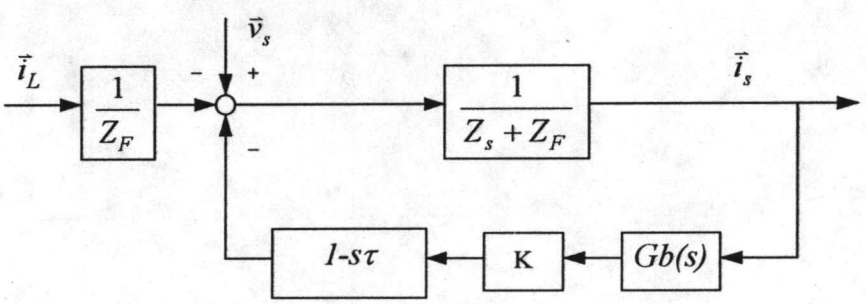
การวิเคราะห์เสถียรภาพระบบเมื่อรวมผลของการประวิงเวลา

จากระบบตามแสดงในรูปที่ 2.10 เราทำการเพิ่มส่วนของการประวิงเวลาอันเนื่องมาจากการคำนวณในไมโครคอนโทรลเลอร์เข้าไปในระบบ โดยในส่วนของการประวิงเวลา ($e^{-s\tau}$) เราจะสามารถประมาณได้โดยอนุกรมเทย์เลอร์ได้คือ

$$e^{-s\tau} \approx 1 - s\tau \qquad (ง.1)$$

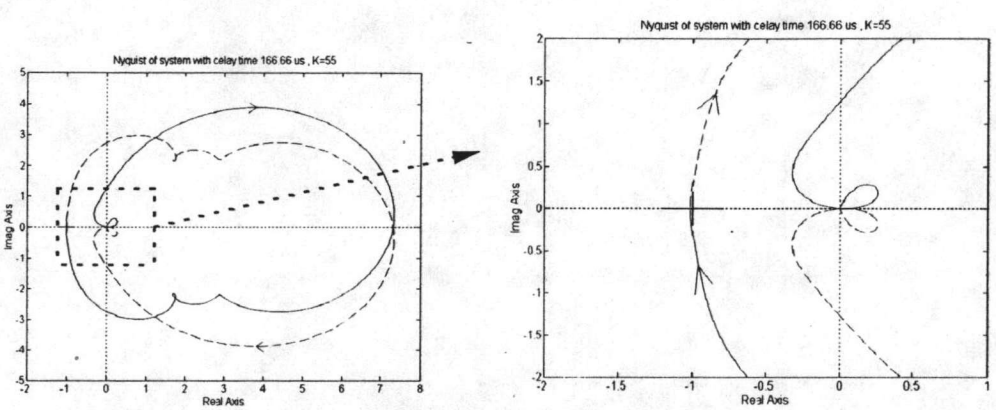
โดย τ คือ เวลาการประวิง

ซึ่งจะทำให้เราสามารถเขียนเขียนระบบใหม่ได้เป็นดังรูปที่ ง.1



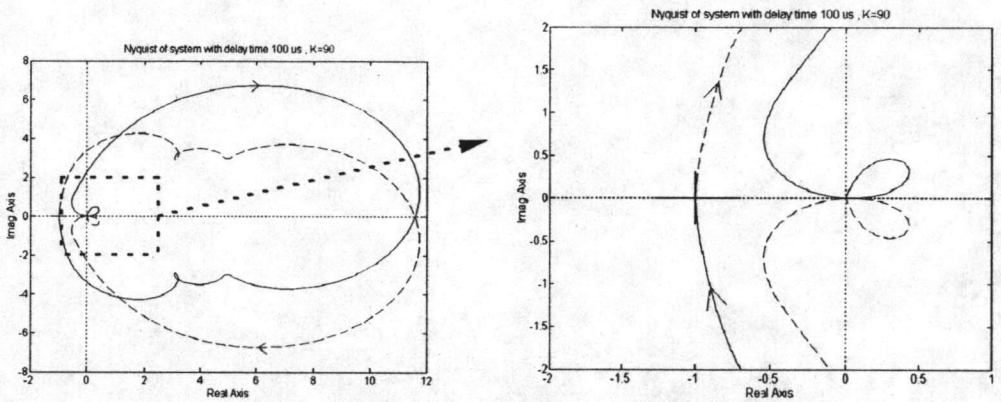
รูปที่ ง.1 บล็อกไดอะแกรมของระบบเมื่อรวมผลของการประวิงเวลา

จากระบบตามรูปที่ ง.1 เมื่อทำการวิเคราะห์ระบบที่มีการประวิงเวลา 166.66 μs โดยแทนค่า τ ในสมการ(ง.1) เป็น 166.66 μs จากนั้นนำไปวิเคราะห์ด้วยแผนภูมิไนควิสต์ ได้ผลตามรูปที่ ง.2 ซึ่งจะเห็นได้ว่าระบบที่มีการประวิงเวลา 166.66 μs จะเริ่มเกิดความไม่เสถียรที่ค่า K ประมาณ 55



รูปที่ ง.2 แผนภูมิไนควิสต์ของระบบที่มีการประวิงเวลา 166.66 μs , K=55

จากนั้นได้ทำการวิเคราะห์ระบบที่มีการประวิงเวลา 100 μs ซึ่งจะได้ผลตามรูปที่ ง.3 จากแผนภูมิไนควิสต์จะเห็นได้ว่าระบบที่มีการประวิงเวลา 166.66 μs จะเริ่มเกิดความไม่เสถียรที่ค่า K ประมาณ 90



รูปที่ ง.3 แผนภูมิไนควิสต์ของระบบที่มีการประวิงเวลา 100 μs , K=90

ภาคผนวก จ

ภาคผนวก จ

ซอฟต์แวร์สำหรับไมโครคอนโทรลเลอร์

ซอฟต์แวร์โมดูล Harmonic

```
;*********************************************************************
;       Program for Harmonic suspression
;*********************************************************************
        .EXPORT   _POWER_RESET_PC,_POWER_RESET_SP
        .EXPORT   _MANUAL_RESET_PC,_MANUAL_RESET_SP
        .EXPORT   _IMIA3,MAIN
        .IMPORT   VAL2HEX,SHOW_R10,SINCOS,INITPWM,AD_IN
        .IMPORT   PWM_GEN,PI_CON,READ_VDC
        .IMPORT   IUV2DQ,FILTER
        .INCLUDE  "MACRO.H"                              ;(MUL,DIV)
        .SECTION MAIN,CODE,ALIGN=4
_POWER_RESET_PC:
_MANUAL_RESET_PC:

MAIN:
        BSR     INITPWM                 ;INITIAL TIMER
        NOP
LOOP:
        BRA     LOOP
        NOP
        .ALIGN 4
;*************************************************************
; INTERRUPT SERVICE
; WHEN TCNT3 = GRA3 THEN INTERRUP
;*************************************************************
_IMIA3:
        STS.L   PR,@-SP
        MOV     TSR_3,R1                ;RESET INTERRUPT
        MOV.B   @R1,R10
        MOV     #0,R10
        MOV.B   R10,@R1
        BSR     READ_IUV                ;READ CURRENT VALUE
        NOP
        MOV     Q,R0
        MOV     @R0,R0
        MOV     IU,R1
        MOV     @R1,R1
        MOV     IV,R2
        MOV     @R2,R2
        BSR     IUV2DQ                  ;CONVERT TO DQ AXIS
        NOP
        MOV     ID,R0
        MOV     R1,@R0
        MOV     IQ,R0
        MOV     R2,@R0
        MOV     IDIN,R0                 ;KEEP ID TO INPUT FILTER
        MOV.W   R1,@R0
        MOV     IQIN,R0
        MOV.W   R2,@R0                  ;KEEP IQ TO INPUT FILTER
        MOV     IDIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Id
```

```
        NOP
        MOV     ID_F,R0
        MOV     R1,@R0
        MOV     IQIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Iq
        NOP
        MOV     IQ_F,R0
        MOV     R1,@R0
LABEL1:BSR      H_FILTER                ;CALCULATE HIGHPASS FILTER
        NOP
        MOV     ID_H,R0                 ;CALCULATE OF COMMAND VOLTAGE
        MOV     @R0,R1                  ;Vc OF d-AXIS
        MOV     IQ_H,R0                 ;Vc OF q-AXIS
        MOV     @R0,R2
        MOV     K_HAR,R3
        MOV     K,R4
        MOV     @R4,R4
        DIVS    R4,R3                   ;R3 = 1/K_HAR
        MOV     R3,R4                   ;R3,R4 = 1/Kharmonic
        DIVS    R3,R1
        MOV     VC_DH,R0
        MOV     R1,@R0                  ;R1 = Vc-d-AXIS(HARMONIC)
        DIVS    R4,R2
        MOV     VC_QH,R0
        MOV     R2,@R0                  ;R2 = Vc-q-AXIS(HARMONIC)
;******************** COMPENSATE DC VOLTAGE *************
DC_COM:
        MOV     R_GAIN,R0
        MOV     @R0,R1
        MOV     ID_F,R0
        MOV     @R0,R2                  ;R2 := ID_F
        MULS    R1,R2                   ;R2 := ID_F * R
        MOV     IQ_F,R0
        STS     MACL,R2
        MOV     I_S,R9
        DIVS    R9,R2
        MOV     @R0,R3                  ;R3 := IQ_F
        MULS    R1,R3                   ;R3 := IQ_F * R
        MOV     VC_DR,R0
        STS     MACL,R3
        MOV     I_S,R9
        DIVS    R9,R3
        MOV     R2,@R0                  ;STORE VC_D RESISTIVE
        MOV     VC_QR,R0
        MOV     R3,@R0                  ;STORE VC_Q RESISTIVE
;*************** COMPENSATE VOLTAGE ***************************
LC_COM:         MOV     X_F,R0
        MOV     @R0,R1                  ;R1 := X
        MOV     #5,R10
        MULS    R10,R1
        MOV     ID_F,R0
        STS     MACL,R1
        MOV     @R0,R2                  ;R2 := ID_F
        DIVS64  R2
        SHAR    R2
        SHAR    R2
        MOV     IQ_F,R0
        MOV     @R0,R3                  ;R3 := IQ_F
        NEG     R3,R4                   ;R4 := -IQ_F
        DIVS64  R4
        SHAR    R4
        SHAR    R4
        MULS    R4,R1
        MOV     VC_DX,R0                ;R3 := VC_DX := - X.IQ_F
        STS     MACL,R3
        DIVS16  R3
        MOV     R3,@R0                  ;COMPENSATE VOLTAGE
        MULS    R2,R1                   ;R4 := VC_QX := X.ID_F
```

```
        MOV     VC_QX,R0
        STS     MACL,R4
        DIVS16  R4
        MOV     R4,@R0                  ;COMPENSATE VOLTAGE
;***************** SUMMING ALL VOLTAGE ******************************
        MOV     VC_DX,R3
        MOV     VC_DH,R0
        MOV     VC_DR,R4
        MOV     VC_D,R5
        MOV     @R0,R1                  ;R1 := VC_DH
        MOV     @R4,R2                  ;R2 := VC_DR
        MOV     @R3,R3                  ;R3 := VC_DX
        ADD     R1,R2                   ;R2 := VC_DH + VC_DR + VC_DX
        ADD     R3,R2
        MOV     R2,@R5                  ;KEEP VC_D
        MOV     VC_QX,R3
        MOV     VC_QH,R0
        MOV     VC_QR,R4
        MOV     VC_Q,R5
        MOV     @R0,R1                  ;R1 := VC_QH
        MOV     @R4,R2                  ;R2 := VC_QR
        MOV     @R3,R3                  ;R3 := VC_QX
        ADD     R1,R2                   ;R2 := VC_QH + VC_QR + VC_QX
        ADD     R3,R2
        MOV     R2,@R5                  ;KEEP VC_Q
OKC:    MOV     Q,R0
        MOV     @R0,R0
        MOV     VC_D,R1
        MOV     @R1,R1
        MOV     VC_Q,R2
        MOV     @R2,R2
        MOV     V_DC,R3
        MOV     @R3,R3
        BSR     PWM_GEN                 ;GENERATE PWM SIGNAL
        NOP
        LDS.L   @SP+,PR
        RTE
        NOP
;*******************************************************
; READ CURRENT PHASE U , V
; FULL SCALE 6 AMP.
; 1 AMP. = 85 dec. = 55 hex.
; OUTPUT = (ANALOG - OFFSET) * GAIN
;*******************************************************
READ_IUV:
        STS.L   PR,@-SP
        MOV     #7,R10                  ;ANALOG READ Iu
        BSR     AD_IN2
        NOP
        ;-------- Do read data from CPU2 during wait ----------------
        ;       Port 1 :  R_GAIN;       Port 2 :  Sign * X_F
        ;-----------------------------------------------------------------
        BSR     INC_Q
        NOP
        MOV     IN1ADR,R0       ;Read R_GAIN
RR_G:   MOV.W   @R0,R1
        MOV.W   @R0,R2
        CMP/EQ  R1,R2
        BF      RR_G            ;Loop read R_GAIN
        MOV     R_GAIN,R0
        MOV     R1,@R0
        MOV     IN2ADR,R0       ;Read S.X_F
R_XF:   MOV.W   @R0,R1
        MOV.W   @R0,R2
        CMP/EQ  R1,R2
        BF      R_XF            ;Loop read S.X_F
        MOV     X_F,R0
        MOV     R1,@R0
        ;---- End Do other thing during wait (1) ---
```

```
        BSR     AD_IN3
        NOP
        MOV     MASK_AD,R11
        AND     R11,R10
        MOV     IU_OFF,R1
        SUB     R1,R10
        MOV     IU_GAIN,R1
        MULS    R1,R10
        MOV     IU,R0
        STS     MACL,R10
        MOV     R10,@R0                         ;KEEP Iu
        MOV     #6,R10                  ;ANALOG READ Iv
        BSR     AD_IN2
        NOP
        ;-------- Do read data during wait ----------
        ;       READ DC VOLTAGE
        ;-------------------------------------------------
        ;BSR    FVC
        ;NOP
        BSR     READ_VDC
        NOP
        MOV     V_DC,R0
        MOV     R1,@R0
        ;---- End Do other thing during wait (2) ---
        BSR     AD_IN3
        NOP
        MOV     MASK_AD,R11
        AND     R11,R10
        MOV     IV_OFF,R1
        SUB     R1,R10
        MOV     IV_GAIN,R1
        MULS    R1,R10
        MOV     IV,R0
        STS     MACL,R10
        MOV     R10,@R0                         ;KEEP Iv
        LDS.L   @SP+,PR
        RTS
        NOP
;*********************************************************
; CALCULATE HIGH PASS FILTER
;*********************************************************
H_FILTER:
        STS.L   PR,@-SP
        MOV     ID_F,R0
        MOV     @R0,R1                  ;R1 = ID OF FUNDAMENTAL
        MOV     IQ_F,R0
        MOV     @R0,R2                  ;R2 = IQ OF FUNDAMENTAL
        MOV     ID,R0
        MOV     @R0,R3                  ;R3 = ID
        MOV     #64,R5
        MULS    R5,R3                   ;CONVERT FROM 10 TO 16 BITS
        MOV     IQ,R0
        MOV     @R0,R4                  ;R4 = IQ
        STS     MACL,R3
        MULS    R5,R4                   ;CONVERT FROM 10 TO 16 BITS
        SUB     R1,R3                   ;R3 = Id-Idf = ID OF HARMONIC
        STS     MACL,R4
        SUB     R2,R4                   ;R4 = Iq-Iqf = IQ OF HARMONIC
        MOV     ID_H,R0                 ;KEEP ID,IQ HARMONIC
        MOV     R3,@R0
        MOV     IQ_H,R0
        MOV     R4,@R0
        LDS.L   @SP+,PR
        RTS
        NOP
        .ALIGN  4
TSR_3:          .DATA.L         H'5FFFF25
IOADR:          .DATA.L         H'93FF000       ;BOARD I/O ADDRESS
OUT1ADR:.       .DATA.L         H'A3FF800       ;EXPANSION OUTPUT PORT 1
```

```
OUT2ADR:        .DATA.L         H'A3FF804       ;EXPANSION OUTPUT PORT 2
IN1ADR:         .DATA.L         H'A3FF810       ;EXPANSION INPUT PORT 1
IN2ADR:         .DATA.L         H'A3FF814       ;EXPANSION INPUT PORT 2
SET_VDC:        .DATA.L         2000            ;(200 * 10)= 1680 VOLT
MIN_VDC:        .DATA.L         500             ;MINIMUM DC VOLTAGE
G_VDC:          .DATA.L          701            ;140.2*5
VDC_OFF:        .DATA.L         49
I_S:            .DATA.L         85*64/2         ;CURRENT SCALE FOR LOOP DC
K_HAR:          .DATA.L         -(85*64/2/10)   ;
K:              .DATA.L         _K
R_GAIN:         .DATA.L         _R_GAIN                 ;GAIN OF RESISTIVE
V_DC:           .DATA.L         _V_DC
V_DCIN:         .DATA.L         _V_DCIN
_V_DCIN:        .DATA.W         0,0,0,0,0,0
_V_DC:          .DATA.L         0
_R_GAIN:        .DATA.L          0
_K:             .DATA.L         65
;*********************************************************
;  INCRESSING OF Q ANGLE
;*********************************************************
INC_Q:  MOV     Q,R1
        MOV     @R1,R0
        ADD     #1,R0
        MOV     MAX_Q,R10
        CMP/EQ  R10,R0
        BF      IGNOR
        MOV     #0,R0
IGNOR:  MOV     R0,@R1                  ;KEEP Q
        RTS
        NOP
;*************************************************************************
; Analog input reading
;*************************************************************************
AD_IN2:
        MOV     R10,R0
        AND     #H'7,R0
        MOV     R0,R11
        AND     #H'3,R0
        SHLL    R0
        MOV     R0,R12
        MOV     R11,R0
        OR      #H'20,R0
        MOV     ADCSR,R13
        MOV.B   R0,@R13
        MOV     TMP,R0
        MOV     R12,@R0
        RTS
        NOP
AD_IN3:MOV      ADCSR,R13
        MOV     TMP,R0
        MOV     @R0,R12
SADINLP0:
        MOV.B   @R13,R0
        TST     #H'80,R0
        BT      SADINLP0
        MOV     ADDR,R0
        MOV.W   @(R0,R12),R10
        SHLR2   R10
        SHLR2   R10
        SHLR2   R10
        RTS
        NOP
;******************** VARIABLE AREA *****************************
        .ALIGN 4
ADCSR:          .DATA.L         H'5FFFEE8
ADDR:
ADDRA:          .DATA.L         H'5FFFEE0
ADDRB:          .DATA.L         H'5FFFEE2
ADDRC:          .DATA.L         H'5FFFEE4
```

```
ADDRD:          .DATA.L         H'5FFFEE6
IV_OFF:         .DATA.L         512-3
IV_GAIN:        .DATA.L         1
IU_OFF:         .DATA.L         512-3
IU_GAIN:        .DATA.L         1
KP:             .DATA.L         1               ;
KI:             .DATA.L         2385            ;KI/16 := 1/(dT/Ti) := Fs * Ti := 6k * 0.1
X_F:            .DATA.L         _X
Q:              .DATA.L         _Q
IU:             .DATA.L         _IU
IV:             .DATA.L         _IV
ID:             .DATA.L         _ID
IQ:             .DATA.L         _IQ
ID_F:           .DATA.L         _ID_F
ID_H:           .DATA.L         _ID_H
IQ_F:           .DATA.L         _IQ_F
IQ_H:           .DATA.L         _IQ_H
VC_D:           .DATA.L         _VC_D
VC_Q:           .DATA.L         _VC_Q
VC_DH:          .DATA.L         _VC_DH
VC_QH:          .DATA.L         _VC_QH
VC_DR:          .DATA.L         _VC_DR
VC_QR:          .DATA.L         _VC_QR
VC_DX:          .DATA.L         _VC_DX
VC_QX:          .DATA.L         _VC_QX
IDIN:           .DATA.L         _IDIN
IQIN:           .DATA.L         _IQIN
AB:             .DATA.L         _AB
OUT:            .DATA.L         _OUT
TMP:            .DATA.L         _TMP
MAX_Q:          .DATA.L         200
_X:             .DATA.L         0
_Q:             .DATA.L         0
_IU:            .DATA.L         0
_IV:            .DATA.L         0
_ID:            .DATA.L         0
_IQ:            .DATA.L         0
_ID_F:          .DATA.L         0
_ID_H:          .DATA.L         0
_IQ_F:          .DATA.L         0
_IQ_H:          .DATA.L         0
_VC_D:          .DATA.L         0
_VC_Q:          .DATA.L         0
_VC_DH:         .DATA.L         0
_VC_QH:         .DATA.L         0
_VC_DR:         .DATA.L         0
_VC_QR:         .DATA.L         0
_VC_DX:         .DATA.L         0
_VC_QX:         .DATA.L         0
_OUT:           .DATA.L         0
_TMP:           .DATA.L         0
_IDIN:          .DATA.W         0,0             ;Idi(k),Idi(k-1)
                .DATA.L         0,0             ;Ido(k-1),Ido(k)
_IQIN:          .DATA.W         0,0             ;Iqi(k),Iqi(k-1)
                .DATA.L         0,0             ;Iqo(k-1),Iqo(k)
_AB:            .DATA.W         32,32           ;b(0),b(1),127,128
;******************** CONSTANT AREA ******************************
MASK_AD:.DATA.L H'3FF
MASK_12:.DATA.L         H'0FFF
MASK_11:.DATA.L         H'07FF
;******************************** wark area *********************************
        .SECTION STACK,STACK,ALIGN=4
        .RES.L 50
_POWER_RESET_SP:
_MANUAL_RESET_SP:
PROG_END:
        .END
```

ซอฟต์แวร์โมดูล Voltage

```
;*********************************************************************
; Program for Voltage compensate
;*********************************************************************
        .EXPORT         _POWER_RESET_PC,_POWER_RESET_SP
        .EXPORT         _MANUAL_RESET_PC,_MANUAL_RESET_SP,_IMIA3
        .IMPORT         SINCOS,INITPWM,AD_IN
        .IMPORT         PWM_GEN,PI_CON,CLR_PWM,VUV2DQ,FILTER,IUV2DQ
        .IMPORT         READ_IUV,SQRT_R10,READ_VD
        .SECTION MAIN,CODE,ALIGN=4
_POWER_RESET_PC:
_MANUAL_RESET_PC:
MAIN:
        BSR     INITPWM                 ;INITIAL TIMER
        NOP
LOOP:
        BRA     LOOP
        NOP
        .ALIGN 4
;*********************************************************
; INTERRUP SERVICE
;*********************************************************
_IMIA3:
        STS.L   PR,@-SP
        MOV     TSR_3,R1                ;RESET INTERRUPT
        MOV.B   @R1,R10                         ; "     "
        MOV     #0,R10                  ; "     "
        MOV.B   R10,@R1                         ; "     "
        BSR     INC_Q                   ;READ Q
        NOP
        BSR     READ_VUV                ;READ LINE VOLTAGE VALUE
        NOP
        BSR     READ_IUV                ;READ CURRENT Iu,Iv
        NOP
        MOV     IU,R0
        MOV     R1,@R0
        MOV     IV,R0
        MOV     R2,@R0
        MOV     QQ,R0
        MOV     @R0,R0
        MOV     VUV,R1
        MOV     @R1,R1
        MOV     VVW,R2
        MOV     @R2,R2
        BSR     VUV2DQ                  ;CONVERT VOLTAGE TO DQ AXIS
        NOP
        MOV     VD,R0
        MOV     R1,@R0
        MOV     VQ,R0
        MOV     R2,@R0
        MOV     VDIN,R0                 ;KEEP VD TO INPUT FILTER
        MOV.W   R1,@R0
        MOV     VQIN,R0
        MOV.W   R2,@R0                  ;KEEP VQ TO INPUT FILTER
        MOV     VDIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Vd
        NOP
        MOV     VD_F,R0                 ;Vs * 64
        MOV     R1,@R0
        MOV     VQIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Vq
        NOP
        MOV     VQ_F,R0                         ;Vs * 64
```

```
        MOV     R1,@R0
        MOV     QQ,R0
        MOV     @R0,R0
        MOV     IU,R1
        MOV     @R1,R1
        MOV     IV,R2
        MOV     @R2,R2
        BSR     IUV2DQ                  ;CONVERT IU,IV TO DQ AXIS
        NOP
        MOV     ID,R0
        MOV     R1,@R0
        MOV     IQ,R0
        MOV     R2,@R0
        MOV     IDIN,R0                 ;KEEP ID TO INPUT FILTER
        MOV.W   R1,@R0
        MOV     IQIN,R0
        MOV.W   R2,@R0                  ;KEEP IQ TO INPUT FILTER
        MOV     IDIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Id
        NOP
        SHAR    R1
        SHAR    R1
        MOV     ID_FX,R0
        MOV     R1,@R0
        DIVS64  R1                      ;DIV TO SCALE
        MOV     ID_F,R0
        MOV     R1,@R0
        MOV     IQIN,R1
        MOV     AB,R2
        BSR     FILTER                  ;LOWPASS FILTER Iq
        NOP
        SHAR    R1
        SHAR    R1
        MOV     IQ_FX,R0
        MOV     R1,@R0
        DIVS64  R1                      ;DIV TO SCALE
        MOV     IQ_F,R0
        MOV     R1,@R0
;******************** COMPENSATE DC VOLTAGE *************
DC_COM:BSR      READ_VDC                ;R6 := V_DC
        NOP
        MOV     V_DC,R0
        MOV     R1,@R0
        MOV     SET_VDC,R6              ;R6 := SET VOLTAGE
        SUB     R6,R1                   ;R1 := PV - SV = ERR VOLTAGE
        MOV     KP,R2                   ;KP = 1
        MOV     KI,R3                   ;KI = 1413
        BSR     PI_CON                  ;COMPENSATE DC VOLTAGE
        NOP                             ;R1 := RESISTIVE GAIN
        MOV     R_GAIN,R0
        MOV     R1,@R0
        MOV     ID_F,R0
        MOV     @R0,R2                  ;R2 := ID_F
        MULS    R1,R2                   ;R2 := ID_F * R
        MOV     IQ_F,R0
        STS     MACL,R2
        MOV     I_S,R9
        DIVS    R9,R2
        MOV     @R0,R3                  ;R3 := IQ_F
        MULS    R1,R3                   ;R3 := IQ_F * R
        MOV     VC_DR,R0
        STS     MACL,R3
        MOV     I_S,R9
        DIVS    R9,R3
        MOV     R2,@R0                  ;STORE VC_D RESISTIVE
        MOV     VC_QR,R0
        MOV     R3,@R0                  ;STORE VC_Q RESISTIVE
        MOV     VD_F,R0                 ; FIND VOLTAGSE SIZE IF VS < 340 THEN EXIT
```

```
        MOV     @R0,R1
        MULS    R1,R1                   ;R3 := Vd^2
        MOV     VQ_F,R0
        MOV     @R0,R2
        STS     MACL,R3
        MULS    R2,R2                   ;R4 := Vq^2
        MOV     VS,R0
        STS     MACL,R4
        ADD     R4,R3
        MOV     R3,@R0                  ;KEEP VS := Vd^2 + Vq^2
        MOV     MIN_VS,R1               ;CHECK Vs > 335
        CMP/GT  R1,R3
        BT      CHK_N
        MOV     #0,R10
        MOV     X,R0
        MOV     R10,@R0
        BRA     FF_X
        NOP
CHK_N:  MOV     MAX_VS,R1               ;CHECK Vs < 380
        CMP/GT  R1,R3
        BF      F_X
        MOV     #0,R10
        MOV     X,R0
        MOV     R10,@R0
        BRA     FF_X
        NOP
        .ALIGN  4
TSR_3:          .DATA.L     H'5FFFF25       ;RESET INTERRUPT
VD:             .DATA.L     _VD
VQ:             .DATA.L     _VQ
QQ:             .DATA.L     _Q
VDIN:           .DATA.L     _VDIN
VQIN:           .DATA.L     _VQIN
IU:             .DATA.L     _IU
IV:             .DATA.L     _IV
ID:             .DATA.L     _ID
IQ:             .DATA.L     _IQ
IDIN:           .DATA.L     _IDIN
IQIN:           .DATA.L     _IQIN
KP:             .DATA.L     1
KI:             .DATA.L     1431            ;KI/16 := 1/(dT/Ti) := Fs * Ti := 10k * Ti/16
SET_VDC:        .DATA.L     2000            ;(200 * 10)= 1680 VOLT
I_S:            .DATA.L     85*64/4/(64*2)  ;CURRENT SCALE FOR LOOP DC
MIN_VS:         .DATA.L     335*335*64*64   ; 335 Volts
MAX_VS:         .DATA.L     380*380*64*64   ; 380 Volts
MIN_VDC:.       DATA.L      500             ;MINIMUM DC VOLTAGE
;*******************************************************
; READ VOLTAGE PHASE U , V FROM ANALOG INPUT
; SCALING 380 Volts to 380 unit by gain 87/256
;*******************************************************
READ_VUV:
        STS.L   PR,@-SP                 ;READ VOLTAGE U-V
        MOV     IN1ADR,R0
        MOV.W   @R0,R6
        SHLL2   R6
        SHLL2   R6
        EXTS.W  R6,R6
        DIVS16  R6                      ;ANALOG VALUE IS 12 BITS
        MOV     #87,R0
        MULS    R0,R6
        STS     MACL,R6
        DIVS16  R6
        DIVS16  R6
        MOV     VUV,R0
        MOV     R6,@R0
        MOV     IN2ADR,R0
        MOV.W   @R0,R6
        SHLL2   R6
        SHLL2   R6
```

```
        EXTS.W R6,R6
        DIVS16 R6
        MOV    #87,R0
        MULS   R0,R6
        STS    MACL,R6
        DIVS16 R6
        DIVS16 R6
        NEG    R6,R6
        MOV    VVW,R0
        MOV    R6,@R0
        LDS.L  @SP+,PR
        RTS
        NOP
        .ALIGN 4
VS:     .DATA.L        _VS
VUV:    .DATA.L        _VUV
VVW:    .DATA.L        _VVW
IN1ADR:.DATA.L         H'A3FF810       ;EXPANSION INPUT PORT 1
IN2ADR:.DATA.L         H'A3FF814       ;EXPANSION INPUT PORT 2
MASK:  .DATA.W         H'0FFF
;================================
; CALCULATE Vs-Vcr
;================================
F_X:    MOV    VD_F1,R6
        MOV    @R6,R1
        MOV    VQ_F1,R7
        MOV    @R7,R2
        MOV    VC_DR,R0
        MOV    @R0,R3
        MOV    VC_QR,R0
        MOV    @R0,R4
        MOV    #51,R5                  ;51/16
        MULS   R5,R3
        STS    MACL,R3
        MULS   R5,R4
        STS    MACL,R4
        DIVS16 R3
        DIVS16 R4
        ADD    R3,R1                   ;Vsd-Vcdr
        ADD    R4,R2                   ;Vsq-Vcqr
        MOV    R1,@R6
        MOV    R2,@R7
;-------------------------------
; - FIND  CUR := Id^2 + Iq^2
;-------------------------------
F_CUR: MOV     ID_F,R0
        MOV    @R0,R1
        MULS   R1,R1                   ;R3 := Id^2
        MOV    IQ_F,R0
        MOV    @R0,R2
        STS    MACL,R3
        MULS   R2,R2                   ;R4 := Iq^2
        MOV    CUR,R0
        STS    MACL,R4
        ADD    R4,R3
        MOV    R3,@R0                  ;KEEP CUR := Id^2 + Iq^2
        MOV    MIN_CUR,R10
        CMP/GT R10,R3
        BT     F_V1                    ;CURRENT > MIN_CURRENT( 1A )
        MOV    #0,R10
        MOV    X,R0
        MOV    R10,@R0
        BRA    FF_X
        NOP
        .ALIGN 4
OUT1ADD:       .DATA.L H'A3FF800
MIN_CUR:       .DATA.L         113     ;0.5 AMP. (0.5*85*64/4/64)^2 = 113
AB:            .DATA.L         _AB
VD_F:          .DATA.L         _VD_F
```

```
VQ_F:           .DATA.L         _VQ_F
;-------------------------------------
; - FIND  V1 := VdId + VqIq
;-------------------------------------
F_V1:   MOV     ID_F,R0
        MOV     @R0,R1                  ;R1 := Id
        MOV     VD_F1,R0
        MOV     @R0,R2                  ;R2 := Vd
        MULS    R1,R2
        MOV     IQ_F,R0
        STS     MACL,R5                 ;R5 := VdId
        MOV     @R0,R3                  ;R3 := Iq
        MOV     VQ_F1,R0
        MOV     @R0,R4                  ;R4 := Vq
        MULS    R3,R4
        MOV     V1,R0                   ;READ ADDRESS V1
        STS     MACL,R6                 ;R6 := VqIq
;-------------------------------------
; - FIND  V2 := abs(VqId - VdIq)
;-------------------------------------
        MULS    R1,R4                   ;VqId
        ADD     R5,R6
        STS     MACL,R1                 ;R1 := VqId
        MULS    R2,R3                   ;VdIq
        MOV     R6,@R0                  ;KEEP V1 := VdId + VqIq
        STS     MACL,R2                 ;R2 := VdIq
        SUB     R2,R1                   ;R1 := VqId - VdIq
        MOV     #1,R2                   ;CHECK SIGN OF V2
        MOV     #0,R0
        CMP/GE  R0,R1                   ;IF V2 > 0 THEN S=-1
        BT      POS
        MOV     #-1,R2
        NEG     R1,R1                   ;If V2 < 0 THEN V2 := -V2
POS:    MOV     SIGN,R0
        MOV.W   R2,@R0
        MOV     V2,R0
        MOV     R1,@R0                  ;KEEP V2 := abs(VqId - VdIq)
;-------------------------------------------------
; - FIND  V1.3 := 380^2/(Id^2 + Iq^2)
;-------------------------------------------------
        MOV     SQR380,R3
        MOV     CUR,R0
        MOV     @R0,R2
        DIVS    R2,R3                   ;R3 := 380^2/(Id^2 + Iq^2)
        MOV     SQG380,R1
        MULS    R1,R3
        STS     MACL,R3
        MOV     V13,R0
        MOV     R3,@R0
;-------------------------------------------------------------
; - FIND  V1.4 := SQR[(VdId + VqIq)/(Id^2 + Iq^2)]
;-------------------------------------------------------------
        MOV     CUR,R0
        MOV     @R0,R2                  ;READ CUR
        MOV     V1,R0
        MOV     @R0,R1
        DIVS    R2,R1                   ;R1 := (VdId + VqIq)/(Id^2 + Iq^2)
        MULS    R1,R1
        STS     MACL,R4                 ;R4 :=SQR[(VdId + VqIq)/(Id^2 + Iq^2)]
        MOV     V14,R0
        MOV     R4,@R0
;-------------------------------------------------------------
; - FIND  V1.5 := SQRT[V1.3-V1.4]
;-------------------------------------------------------------
        SUB     R4,R3
        MOV     R3,R10
        BSR     SQRT_R10
        NOP                             ;R10 := SQRT[V1.3-V1.4]
        MOV     V15,R0
```

```
        MOV     R10,@R0
;-------------------------------------------------------
; - FIND  V2.1 := abs(VqId - VdIq)/(Id^2 + Iq^2)
;-------------------------------------------------------
        MOV     V2,R0
        MOV     @R0,R1
        MOV     CUR,R0
        MOV     @R0,R2
        DIVS    R2,R1                   ;R1 := (VdId + VqIq)/(Id^2 + Iq^2)
        MOV     V21,R0
        MOV     R1,@R0
;---------------------------------
; - FIND  X := V1.5 - V2.1
;---------------------------------
        SUB     R1,R10
        MOV     X,R0
        MOV     R10,@R0                 ;KEEP  X
FF_X:   MOV     R10,R1
        MOV     X_F,R0
        MOV     R1,@R0
;***************** SEND DATA TO CPU 1 ******************************
;       Port 1 :  R_GAIN;          Port 2 :  Sign * X_F
;*********************************************************************
CPU1:
        MOV     R_GAIN,R0               ;MOVE R_GAIN TO CPU 1
        MOV     @R0,R1
        MOV     OUT1ADR,R0
        MOV.W   R1,@R0
        MOV     X_F,R0                  ;MOVE X GAIN TO CPU 1
        MOV     @R0,R1
        MOV     SIGN,R0
        MOV.W   @R0,R2
        MULS    R1,R2
        STS     MACL,R1
        MOV     OUT2ADR,R0
        MOV.W   R1,@R0
;******************** End Send data to CPU 1 **************************
        LDS.L   @SP+,PR
        RTE
        NOP
        .ALIGN  4
R_GAIN:         .DATA.L         _R_GAIN                 ;GAIN OF RESISTIVE
AB1:            .DATA.L         _AB
VD_F1:          .DATA.L         _VD_F
VQ_F1:          .DATA.L         _VQ_F
ID_F:           .DATA.L         _ID_F
IQ_F:           .DATA.L         _IQ_F
ID_FX:          .DATA.L         _ID_FX
IQ_FX:          .DATA.L         _IQ_FX
XIN:            .DATA.L         _XIN
V_DC:           .DATA.L         _V_DC
V13:            .DATA.L         _V13
V14:            .DATA.L         _V14
V15:            .DATA.L         _V15
V21:            .DATA.L         _V21
;*********************************************************
;  INCRESSING OF Q ANGLE
;*********************************************************
INC_Q:  MOV     Q,R1
        MOV     @R1,R0
        ADD     #1,R0
        CMP/EQ  #120,R0
        BF      IGNOR
        MOV     #0,R0
IGNOR:  MOV     R0,@R1                  ;KEEP Q
        RTS
        NOP
Q:              .DATA.L         _Q
SQR380:         .DATA.L         381*381         ;(380)^2
```

```
SQG380:         .DATA.L         64*64           ;GAIN OF VOLTAGE
DIV_V:          .DATA.L         65536
X:              .DATA.L         _X
X_F:            .DATA.L         _X_F
VC_D:           .DATA.L         _VC_D
VC_Q:           .DATA.L         _VC_Q
VC_DH:          .DATA.L         _VC_DH
VC_QH:          .DATA.L         _VC_QH
VC_DR:          .DATA.L         _VC_DR
VC_QR:          .DATA.L         _VC_QR
VC_DX:          .DATA.L         _VC_DX
VC_QX:          .DATA.L         _VC_QX
CUR:            .DATA.L         _CUR
V1:             .DATA.L         _V1
V2:             DATA.L          _V2
_ID_FX:         .DATA.L         0
_IQ_FX:         .DATA.L         0
SIGN:           DATA.L          _SIGN
OUT1:           .DATA.L         _OUT1
OUT2:           .DATA.L         _OUT2
_R_GAIN:        .DATA.L         0
_ID:            .DATA.L         0
_IQ:            .DATA.L         0
_X_F:           .DATA.L         0
_Q:             .DATA.L         0
_IU:            .DATA.L         0
_IV:            .DATA.L         0
_VUV:           .DATA.L         0
_VVW:           .DATA.L         0
_VS:            .DATA.L         0
_V_DC:          .DATA.L         0
_VD:            .DATA.L         0
_VQ:            .DATA.L         0
_VC_D:          .DATA.L         0
_VC_Q:          .DATA.L         0
_VC_DH:         .DATA.L         0
_VC_QH:         .DATA.L         0
_VC_DR:         .DATA.L         0
_VC_QR:         .DATA.L         0
_VC_DX:         .DATA.L         0
_VC_QX:         .DATA.L         0
_VD_F:          .DATA.L         1               ;OUTPUT VARIABLE START HERE
_VQ_F:          .DATA.L         2
_ID_F:          .DATA.L         3
_IQ_F:          .DATA.L         4
_CUR:           .DATA.L         5
_V1:            .DATA.L         6
_V2:            .DATA.L         7
_V13:           .DATA.L         8
_V14:           .DATA.L         9
_V15:           .DATA.L         10
_V21:           .DATA.L         11
_X:             .DATA.L         12
_VDIN:  .DATA.W         0,0,0,0,0,0
_VQIN:  .DATA.W         0,0,0,0,0,0
_IDIN:  .DATA.W         0,0,0,0,0,0
_IQIN:  .DATA.W         0,0,0,0,0,0
_XIN:   .DATA.W         0,0,0,0,0,0
_AB:    .DATA.W         32,32
_SIGN:  .DATA.W         0
;******************** CONSTANT AREA *****************************
        .INCLUDE "ADDRESS.H"
;******************************** work area *********************************
        .SECTION STACK,STACK,ALIGN=4
        .RES.L 50
_POWER_RESET_SP:
_MANUAL_RESET_SP:
PROG_END:
        .END
```

ประวัติผู้เขียน

นายกิติพจน์ สิทธิเลิศพิศาล เกิดเมื่อวันที่ 10 พฤษภาคม พ.ศ. 2511 ที่เขตบางกอกน้อย กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า (ไฟฟ้ากำลัง) จากมหาวิทยาลัยเชียงใหม่ ปีการศึกษา 2532 และ ได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า (อิเล็กทรอนิกส์กำลัง) ณ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535