19

Extended Simulation of an Embedded Brushless Motor Drive (BLMD) System for Adjustable Speed Control Inclusive of a Novel Impedance Angle Compensation Technique for Improved Torque Control in Electric Vehicle Propulsion Systems

Richard A. Guinee
Cork Institute of Technology
Ireland

1. Introduction

As already stated for the reasons given in a previous chapter a good continuous time model, of low complexity, of a BLMD system is essential to adequately describe mathematically the PWM inverter switching process with dead time and subsequent binary waveform generation in terms of the switching instant occurrences for accurate computer aided design (CAD) of embedded BLMD model simulation in proposed electric vehicle (EV) propulsion systems. In this chapter a complete software model of the BLMD system as a set of difference equations representing subsystem functionality, the organization of these subsystem activities into flowchart form and the processing details of these modular activities as software function calls in C-language for simulation purposes (Guinee, 2003) is presented.

Furthermore in this the second chapter, concerning BLMD model fidelity for EV applications, BLMD model simulation accuracy for embedded EV CAD is next checked for a range of restraining shaft load torques via numerical simulation and then extensively compared and benchmarked for accuracy against theoretical estimates using known manufacturer's catalogued specifications and motor drive constants (Guinee, 2003).

Model simulation accuracy is further substantiated and validated through evaluation of the shaft velocity step response rise time when cross checked against (i) experimental test data and (ii) that evaluated from the catalogued performance index relating to the brushless motor dynamic factor (Guinee, 2003). Numerical simulation with outer velocity loop closure is used to demonstrate the accuracy of the completed BLMD reference model, based on established model confidence in torque control mode, in ASD configuration when compared with experimental test data.

In addition to the BLMD model structure presented in the previous chapter for actual drive emulation two innovative measures which relate to increased drive performance are also provided. These novel techniques (Guinee, 2003), which include

i. inverter dead time cancellation and
ii. motor stator winding impedance angle compensation,

are encapsulated within the BLMD model framework and simulated for validation purposes and prediction of enhanced drive performance in EV systems. An approximate analysis is given to support the approach taken and verify the performance outcome in each case.

In the first of these BLMD performance enhancements a novel compensation method has already been presented in the first chapter to offset the torque reduction effects of inverter delay during BLMD operation. This simple expedient relies on the zener diode clamping of the triangular carrier voltage during the carrier waveform comparison with the modulating current control signal in the comparator modulator to nullify power transistor turnon delay. This approach obviates the need for separate compensation timing circuitry in each phase as required in other schemes. The accuracy of this methodology is supported by current feedback, EM torque generation and shaft velocity trace simulation when compared with similar traces from the BLMD benchmark reference model with the effects of the inverter basedrive trigger delay neglected.

The second proposed innovative improvement, presented in this chapter, relates to the progressive introduction of commutation phase lead with increased shaft speed as BLMD impedance angle compensation which forces the impedance angle to the same value as the internal power factor angle. This effect maintains zero load angle between the stator winding terminal voltage and the back emf. It also results in rated load torque delivery at lower shaft speeds with minimal rise time, overshoot and settling time in the generated torque for a range of torque demand input values. This novel technique greatly enhances the dynamic performance of the embedded BLMD prime mover in EV applications without overstressing mechanical assembly components during periods of rapid acceleration and deceleration. The incorporation of this novel impedance angle compensation technique thus minimizes component wear-out such as gear boxes, transmission shafts and wheel velocity joints and consequently enhances overall EV reliability improvement. BLMD simulation is provided in torque control mode at rated torque load conditions, for the actual drive system represented, with and without impedance angle compensation to gauge model performance accuracy over a range of torque demand step input values.

2. BLMD model structure and program sequence of activities

The BLMD model structure is composed of interconnected subsystems with feedback as shown in Figure 1, of varying complexity according to physical principles. Consequently it can be described by a discrete time configuration of first order digital filter realizations for linear elements cascaded with difference equations representing nonlinear PWM inverter behaviour into a complete software model for simulation purposes as illustrated in Figure 2. The BLMD model program is organized into a sequence of software activities, coded in C-language as function calls, representing the functionality of various subsystem modules shown as the flowchart in Figure 3. All subsystem output (o/p) variable quantities in the cascaded activity chain are assumed to remain constant, once computed irrespective of feedback linkage, throughout the remainder of the time step interval Δt_k based on the simulation sampling rates $(1/\Delta t_k)$ chosen from considerations given in section 3.1 of the previous chapter. The essential features of the BLMD model program in Figure 3 can be explained by means of the linked modular software configuration encoded as the functional block sequence in Figure 2 along with the appropriate C-language code segments illustrated in Figure 4.

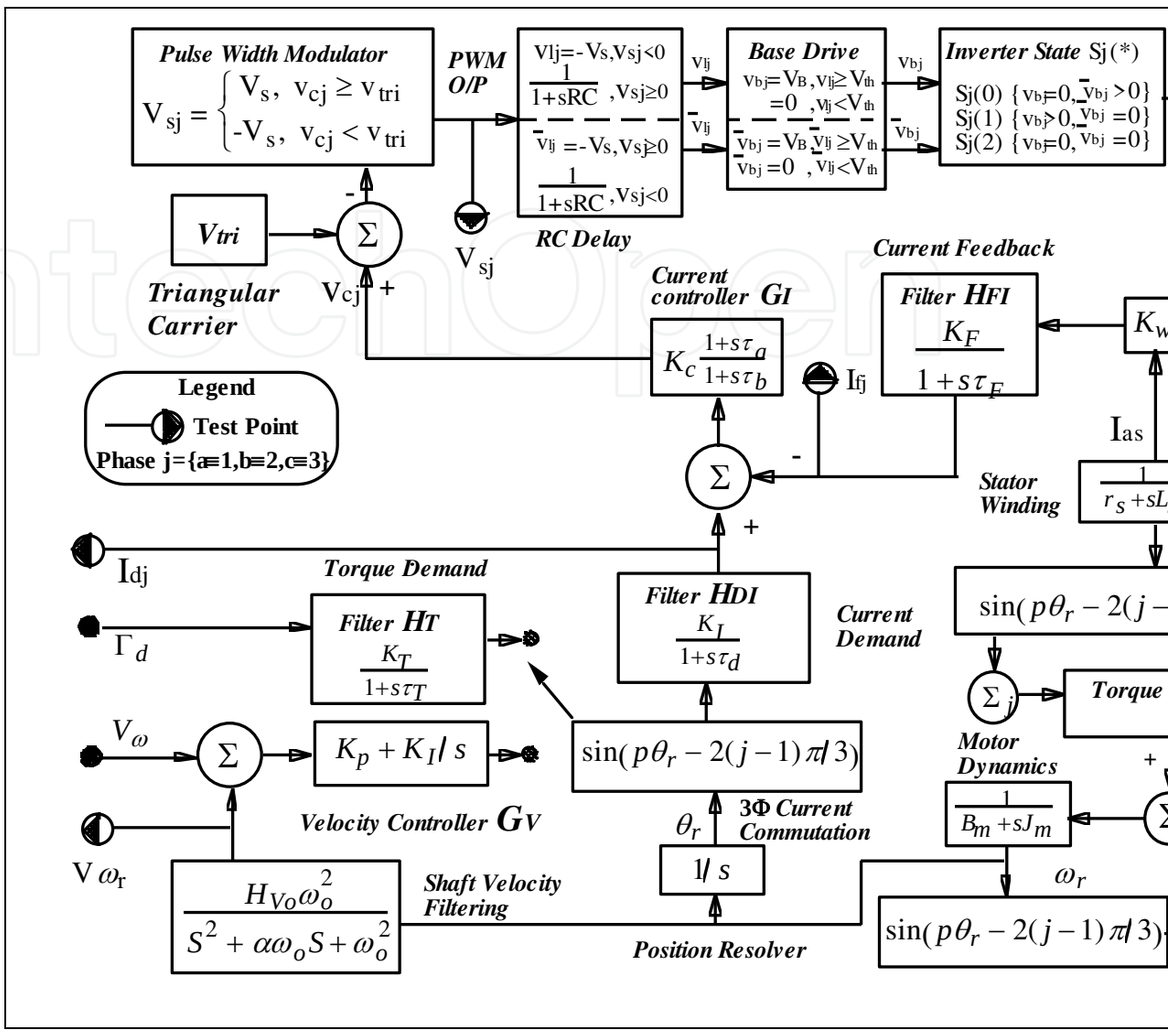


Fig. 1. Transfer Function Block Diagram of a BLMD System (Guinee, 1999)

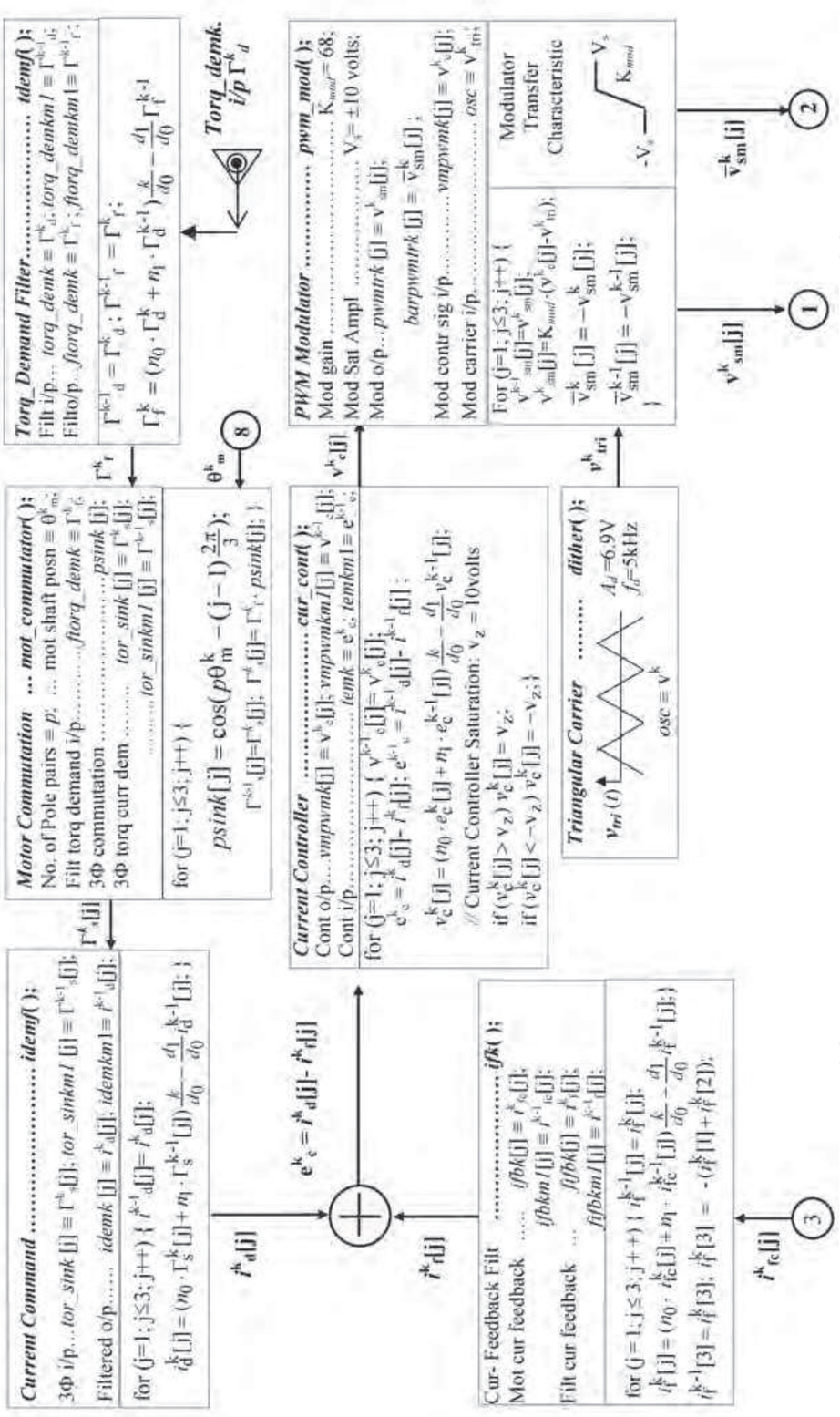


Fig. 2-A. Software Functional Block Diagram (Guinee, 2003) of a BLMD System

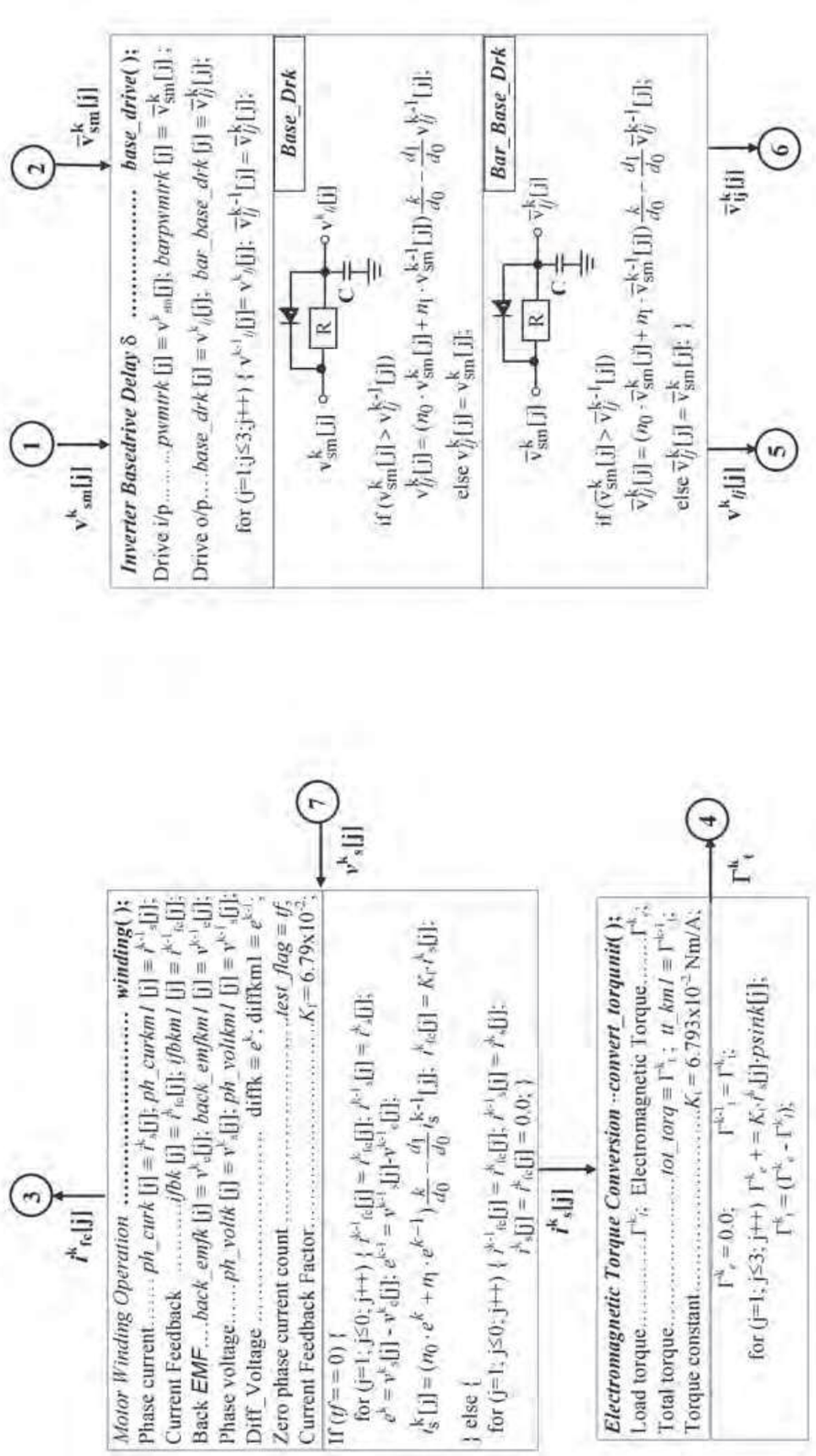


Fig. 2-B. Software Functional Block Diagram of a BLMD System

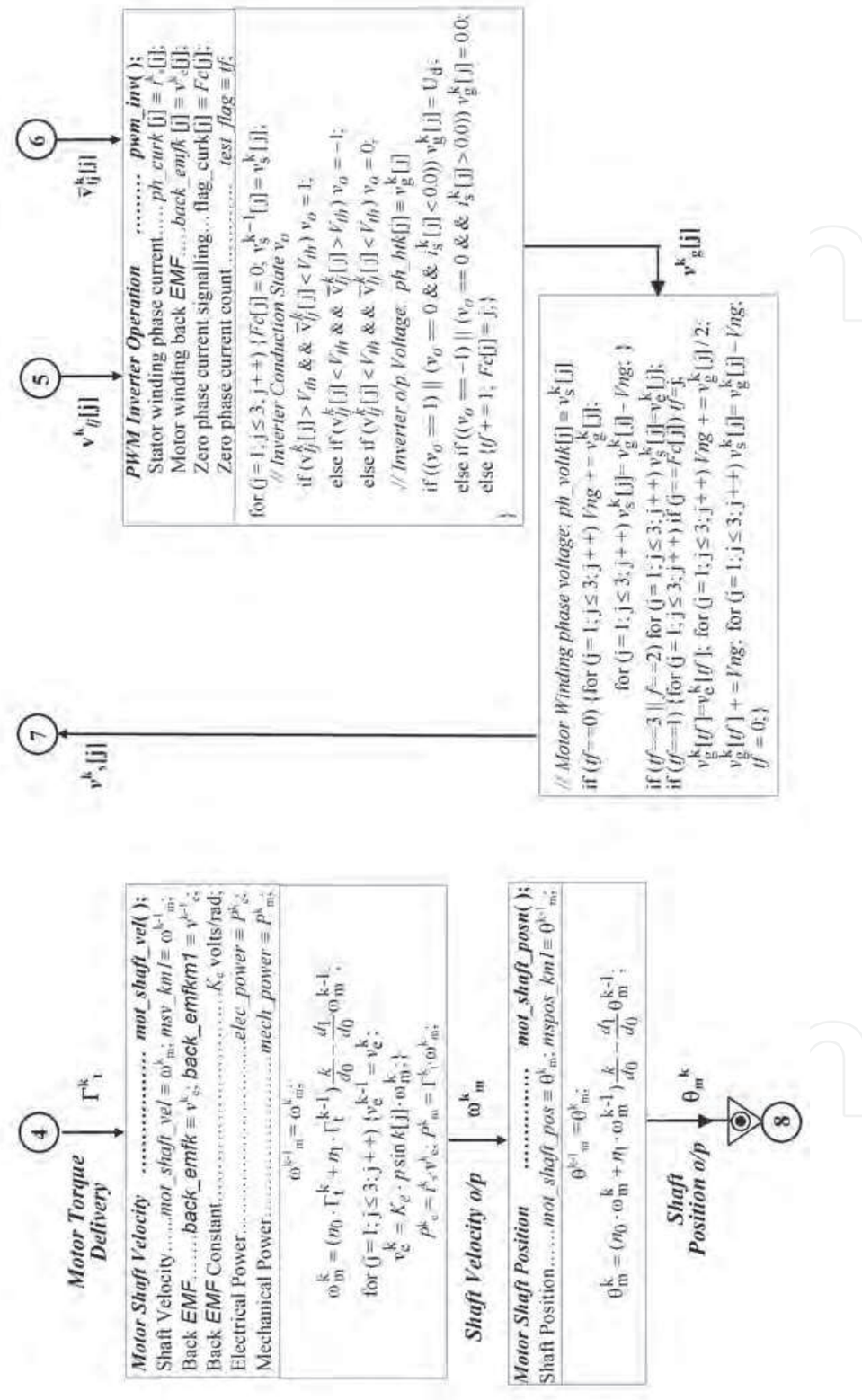


Fig. 2-C. Software Functional Block Diagram of a BLMD System

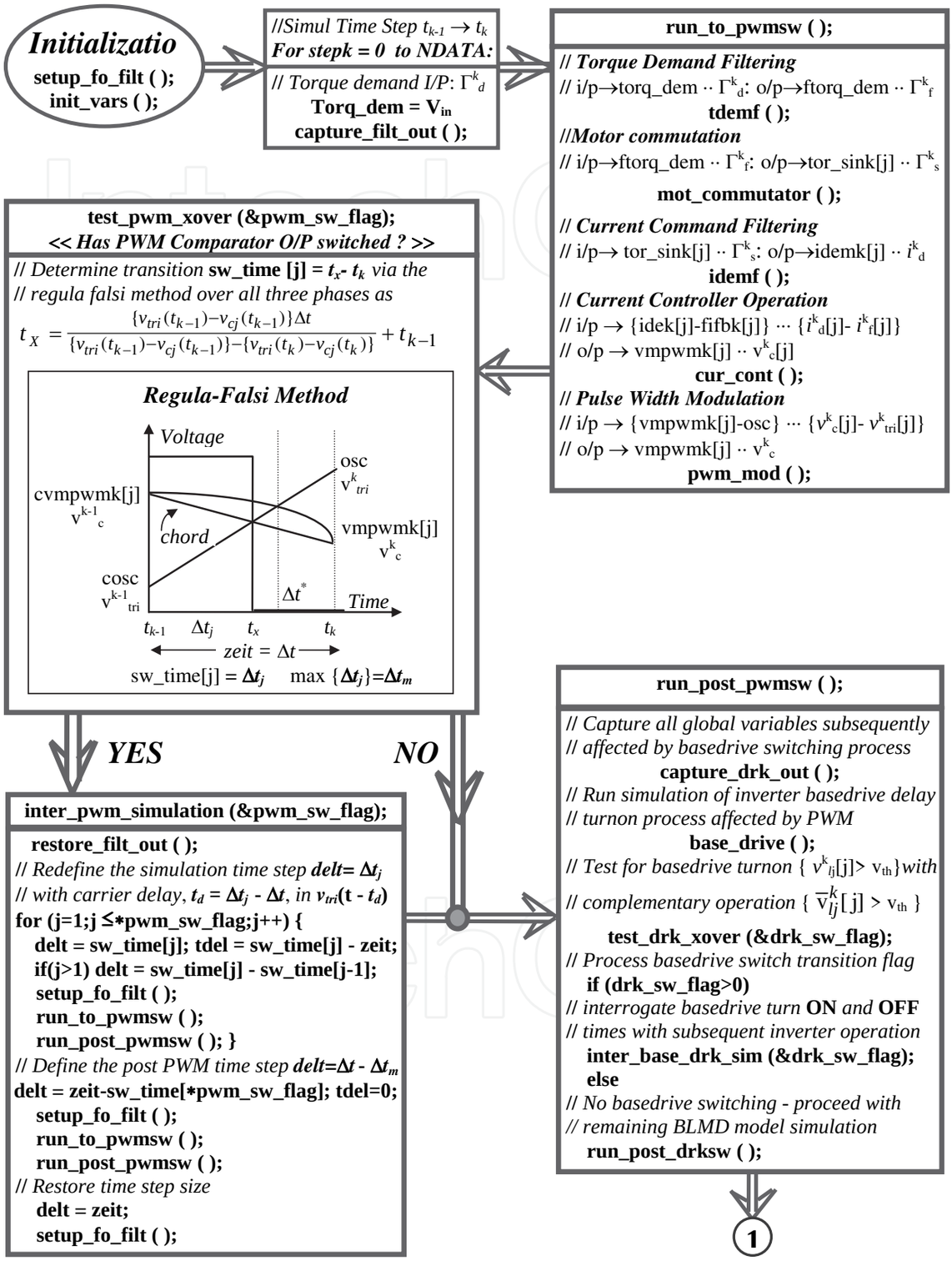


Fig. 3-A. Program Flow Diagram (Guinee, 2003) for BLMD Model Simulation

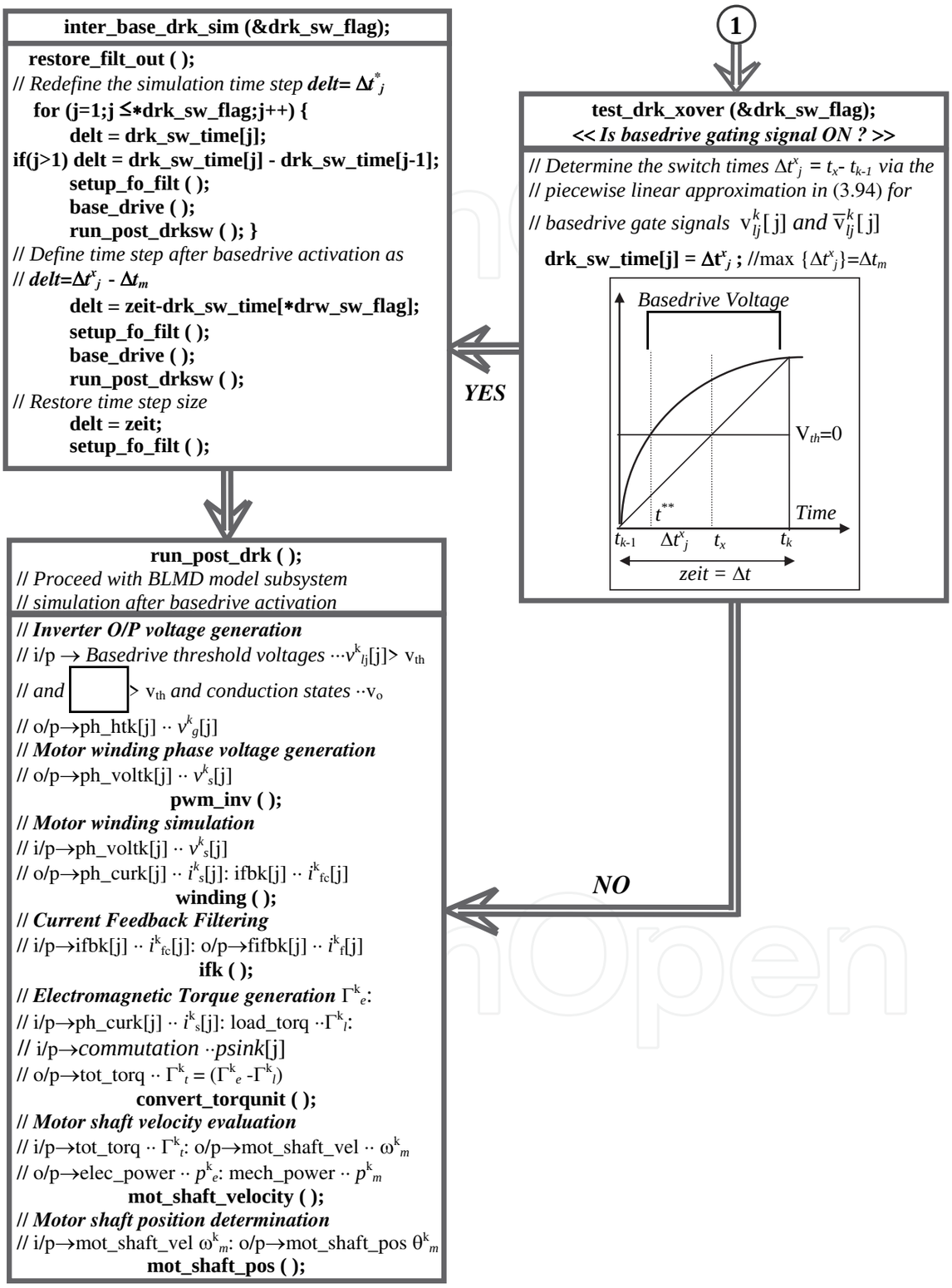


Fig. 3-B. Program Flow Diagram for BLMD Model Simulation

2.1 BLMD model simulation

Numerical simulation commences with the declaration of known BLMD system parameters followed by a declaration with initialization of variables and three phase (3Φ) arrays for global usage, over the program linked function call sequence, as outlined in the code blocks shown in Figure 4-1.

≈ Define BLMD System Parameters in Fig.5 ≈

// ***Define all filter constants*:**

K_T=1.0; τ_T =222μS; // ***Torque Demand Filter*** H_T

K_I=1.0; τ_I =100μS; // ***Current Demand Filter*** H_{DI}

K_F =5.0; τ_I =47μS; //***Current Feedback Filter*** H_{FI}

$K_{wi} \equiv K_f$; // ***Current Feedback Factor***

K_C =19.5; τ_a =225μS; τ_b=1.5mS; //***Current Controller*** G_I

H_{vo}=13.5x10^{-3}; α=√2; ω_o=2x10^3 rad.s^{-1}; //***Velocity Filter*** H_V

K_d=1.0; τ_d =28.6μS; // ***RC Delay***

r_S=0.75Ω; L_S=1.94mH; p = 6 pole pairs; // ***Motor Winding***

K_e = K_t =0.3 // ***Motor torque & Back EMF Constants***

J_m=3 kg.cm^2; B_m=2.14x10^{-3} Nm.rad^{-1}; // ***Motor Dynamics***

U_d=310 Volts; V_{th}=0; V_S=10 Volts; // ***Voltage Constants***

f_S =5.7kHz; A_d=6.9 Volts; // ***Carrier Waveform Constants***

≈ Define Global Variables ≈

stepk // **kth** ***iteration time step***

delta_t // ***Fixed simulation Time Step Size*** $\Delta t = (t_k - t_{k-1})$

delt // ***Variable simulation Time Step Size*** Δt_i

tdel // ***Carrier*** $v_{tri}(t-t_d)$ ***time delay*** t_d

osc // ***Carrier amplitude*** v^k_{tri}

torq_dem // ***motor i/p torque demand*** Γ^k_d

ftorq_dem // ***Filtered torque demand*** Γ^k_f

mot_shaft_pos // ***Motor shaft position*** θ^k_m

mot_shaft_vel // ***Motor shaft velocity*** ω^k_m

tot_torq // ***Net drive torque*** Γ^k_f

≈ Define Global Array variables for j=1 to3 ≈

ppsinkm1[j] // ***3-phase commutation psink[j]***

tor_sink[j] // ***3-phase torque demand*** Γ^k_s**[j]**

idemk[j] // ***3-phase current demand*** i^k_d**[j]**

fifbk[j] //***Filtered current feedback*** i^k_f**[j]**

ifbk[j] // ***Current feedback*** i^k_{fc}**[j]**

vmpwmk[j] // ***Current controller o/p*** v^k_c**[j]**

pwmtrk[j] // ***Modulator o/p*** v^k_{sm}**[j]**

base_drk[j] // ***Basedrive o/p*** v^k_{lj}**[j]**

bar_base_drk[j] // ***Complementary basedrive o/p*** $\bar{v}^k_{lj}$[j]

ph_voltk[j] // ***Stator winding Phase voltage*** v^k_s**[j]**

back_emfk[j] // ***Motor back EMF*** v^k_e**[j]**

ph_curk[j] // ***Stator winding Phase current*** i^k_s**[j]**

≈ init_vars ( ) ≈

// ***Function initializes all global variables and arrays to*** **0.0**

Fig. 4-1. Declaration and Initialization (Guinee, 2003)

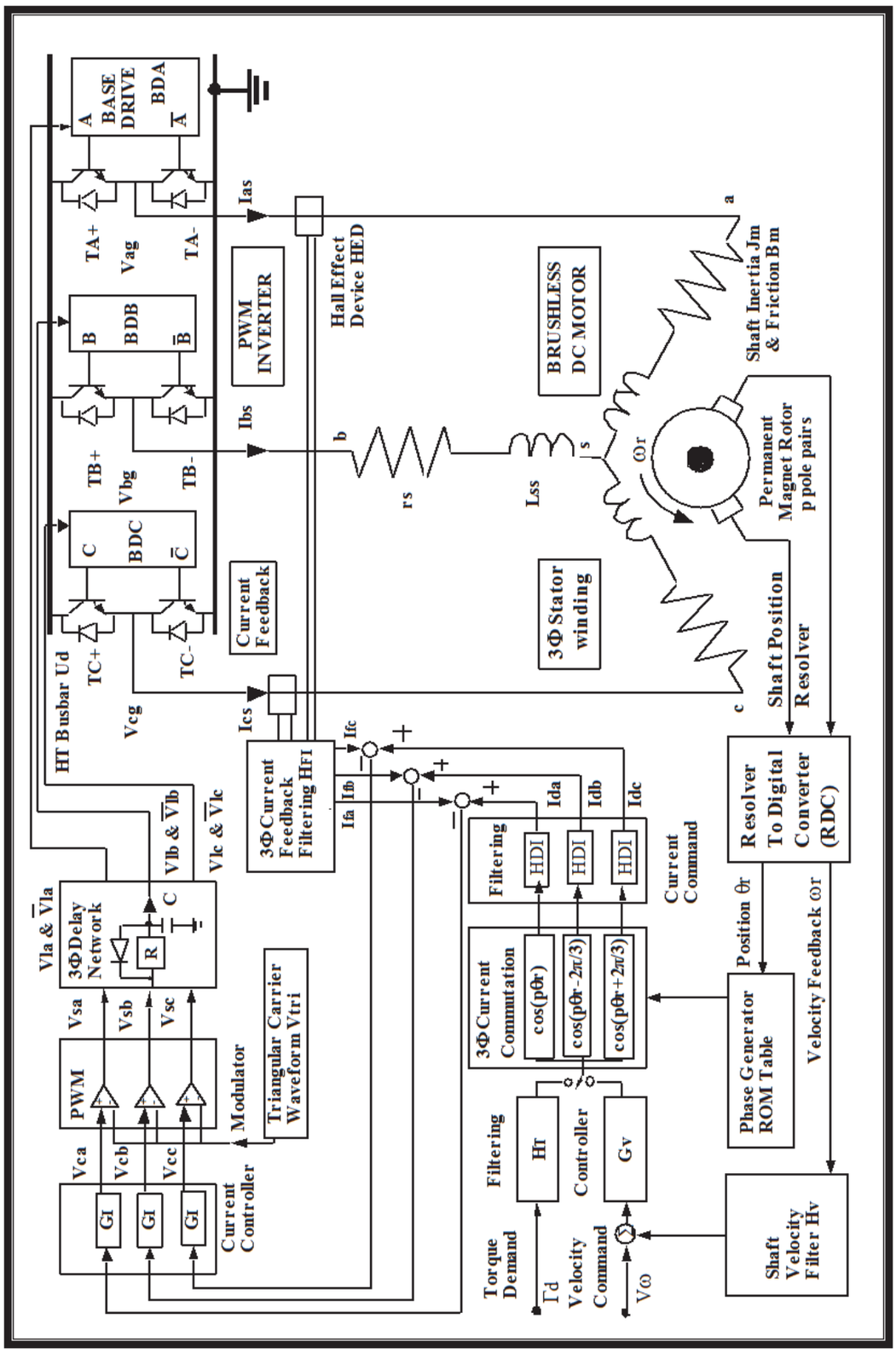


Fig. 5. Network Structure (Guinee, 1999) of a Typical Brushless Motor Drive System

All first order linear system discretization is accomplished by complex variable substitution of the Euler backward rectangular rule using the Z transform. The alternative filter discretization process using Tustin's bilinear method (Franklin et al, 1980) or the trapezoidal integration rule (Balabanian, 1969) can also be used but with negligible observable differences at the step size Δt chosen. Resultant digital filter implementation for simulation purposes is facilitated by transfer of the appropriate filter time constant and gain coefficients using the C-language 'structure' mechanism in the function call *setup_fo_filt* **( )** illustrated in Figure 4-2.

≈ *setup_fo_filt* ( ) ≈

// ***Function sets up and discretizes all first order BLMD***
// ***linear subsystems H(s)*** in Figures 1 & 5 according
// to expressions (LX) and (LXIV) in the previous chapter as

$$H(s) = K_c \frac{1+\tau_a s}{1+\tau_b s} \xrightarrow{s \sim \frac{1}{T}(1-z^{-1})} k \frac{n_0 + n_1 z^{-1}}{d_0 + d_1 z^{-1}}$$

// using function call ***discrete1*( );**
// with discrete filter coefficients $\{k, n_0, n_1, d_0, n_1\}$, such as
***tdemf*( ); *idemf*( ); *ifk*( ); *cur_cont*( ); *winding*( );**
***mot_shaft_vel*( ); *mot_shaft_posn*( );**
***dtrdfilt*( );** // drive transistor RC delay network

≈ First Order System Discretization Process *discrete1* ( ) ≈

```
typedef struct
{
      double k;
      double n0;
      double n1;
      double d0;
      double d1;
} tf1;
```

// ***Tustin's Bilinear Method***

// $s \sim \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$

```
tf1 discrete1(tf1 temp)
{
   tf1 discrete;
   double at,bt,ts;
   double k,n0,n1,d0,d1;
   ts=delt;
   bt=temp.d0*ts/2.0;
   at=temp.n0*ts/2.0;
discrete.n0=at+temp.n1;
discrete.n1=at-temp.n1;
discrete.d0=bt+temp.d1;
discrete.d1=bt-temp.d1;
 discrete.k=temp.k;
   return discrete;}
```

// ***Euler's Backward Integration Rule***

// $s \sim \frac{1}{T}(1-z^{-1})$

```
double delt;
tf1 discrete1(tf1 temp)
{
      tf1 discrete;
      double at,bt,ts;
      double k,n0,n1,d0,d1;
      ts=delt; k=temp.k;
      n0=temp.n0; n1=temp.n1;
      d0=temp.d0; d1=temp.d1;
      at=ts*n0; bt=ts*d0;
      n0=at+n1; d0=bt+d1;
      if(n1!=0.0) {
discrete.k=k*n1/d1;
discrete.n0=n0/n1;discrete.n1 = -1.0;
discrete.d0=d0/d1; discrete.d1 = -1.0;
              } else {
discrete.k=k*n0/d1;
discrete.n0=1.0;
discrete.n1 = 0.0;
discrete.d0=d0/d1;
discrete.d1 = -1.0;
}
return discrete; }
```

Fig. 4-2. Linear Subsystem Discretization

Before proceeding with model program execution in the k^{th} time step ($t_{k-1} \rightarrow t_k$) all global variables and arrays are captured in the function call *capture_filt_out* **( )** for later reinstatement, during accurate resolution of the width modulated pulse edge transition time via the regula-falsi method, with **restore_filt_out ( )** in Figure 4-3.

```
≈ capture_filt_out ( ) ≈
// Capture all global variables and arrays for PWM evaluation
void capture_filt_out(void)
{
   int j;
   cftorq_dem=ftorq_dem; cosc=osc;
   ctot_torq=tot_torq; cmot_shaft_vel=mot_shaft_vel;
   cmot_shaft_pos=mot_shaft_pos;
   for(j=1;j<=3;j++) {
      cidemk[j]=idemk[j]; ctor_sink[j]=tor_sink[j];
      cvmpwmk[j]=vmpwmk[j]; cpwmtrk[j]=pwmtrk[j];}
    return;}
```

```
≈ restore_filt_out ( ) ≈
/ /Function reinstates  arrays and global variables
    ftorq_dem = cftorq_dem; //etc. for all other variables
    for (j=1;j≤3; j++) {
          idemk[j] = cidemk[j]; tor_sink[j]=ctor_sink[j];
          vmpwmk[j]=cvmpwmk[j]; pwmtrk[j]=cpwmtrk[j];
    }
```

Fig. 4-3. Variable capture and restoration

The instruction code group *run_to_pwmsw* ( ) processes the sequence of BLMD software activities up to the comparator modulator o/p using the following list of function calls in Figures 4-1 and 4-4.

tdemf ( ) filters the i/p torque demand signal *torq_dem* with o/p *ftorq_dem*.

mot_commutator ( ) establishes the 3-phase reference *psink***[j]**, from the computed shaft rotor displacement θ^{k}_{m}, for 3 phase stator winding voltage commutation with modulated amplitude.

tor_sink[j] based on the filtered torque demand *ftorq_dem*.

idemf ( ) filters the i/p torque related current command signal *tor_sink***[j]** with o/p *idemk***[j]**.

cur_cont ( ) The lag compensator 'optimizes' the current error as the difference between the current command *idemk***[j]** and the filtered stator winding current feedback *fifbk*[j] in each phase of the 3Φ current control loop. The o/p *vmpwmk***[j]** from each of the high gain controllers is amplitude limited to the saturation voltage levels $\pm V_z$ (~10v) by zener diodes.

pwm_mod ( ) produces a width modulated o/p pulse sequence *pwmtrk***[j]** for each phase in accordance with the amplitude comparison of the modulating control signal *vmpwmk*[j] o/p and the triangular dither signal *osc*. The modulator has a gain K_{mod} with complementary outputs, for basedrive operation, that are hard limited to $\pm V_z$ (~10v) by zener diodes.

≈ *run_to_pwmsw* ( ) ≈

≈ *tdemf* ( ) ≈ 1

```
// I/P Torque demand filtering
double torq_dem,td_km1;
double ftorq_dem,ftd_km1;
void tdemf(void)
{
ftd_km1= ftorq_dem;
ftorq_dem = fil_torq_sign*(dtdfilt.n0*torq_dem
          +dtdfilt.n1*td_km1)*dtdfilt.k/dtdfilt.d0
          -dtdfilt.d1*ftd_km1/dtdfilt.d0;
return; }
```

≈ *mot_commutation* ( ) ≈ 2

```
// Three phase current commutation
double mot_shaft_pos; // Motor shaft position
void mot_commutator(void)
{
  int j;
  double temp1,temp2;
  temp1=NPOLE*mot_shaft_pos;
  temp2=2*PI/3;
  ppsink[1]= cos(temp1);
  ppsink[2]= cos(temp1-temp2);
  ppsink[3] = -(ppsink[1]+ppsink[2]);
  for(j=1;j<=3;j++) {tor_sinkm1[j]=tor_sink[j];
  tor_sink[j]=ppsink[j]*ftorq_dem;}
  return;}
```

≈ *idemf* ( ) ≈ 3

```
// Current command filtering
void idemf(void)
{ int i;
          for (i=1;i<=3;i++) {
          idemkm1[i]=idemk[i];
idemk[i]=(didfilt.n0*tor_sink[i]
 +didfilt.n1*tor_sinkm1[i])*didfilt.k/didfilt.d0
 -didfilt.d1*idemkm1[i]/didfilt.d0; }
          return;}
```

≈ *cur_cont* ( ) ≈ 4

```
// BLMD Current controller simulation
double cont_errk;
void cur_cont(void)
{
 int i;
 double tempk,tempkm1;
 cont_errk=0.0;
 for(i=1;i<=3;i++) {
 vmpwmkm1[i]=vmpwmk[i];
 tempk=idemk[i]-fifbk[i];
 tempkm1=cont_errk = idemkm1[i]-fifbkm1[i];
 vmpwmk[i]= (dicont.n0*tempk
     +dicont.n1*tempkm1)*dicont.k/dicont.d0
      - dicont.d1*vmpwmkm1[i]/dicont.d0;
if(vmpwmk[i]>Vz+Vd) {vmpwmk[i]=Vz+Vd;
else
if(vmpwmk[i] < -(Vz+Vd)) {vmpwmk[i] = -
                    (Vz+Vd);}
return;}
```

Fig. 4-4. Call Sequence to PWM O/P

A test is used to interrogate the o/p status of the simulated comparator modulator by monitoring any observational sign change in the o/p polarity ($\pm V_Z$), which is indicative of a modulated pulse edge transition, after execution of the software code module *pwm_mod* **( )** as shown in Figure 4-5.

≈ *pwm_mod* ( ) ≈ 5

```
// Pulse Width Modulator Function
double Vd=0.0; // RC shunt diode volt drop
double Vz=10.0; // Modulator saturation limits
double osc; // Carrier amplitude
void pwm_mod(void) {
 double modop; // Modulator o/p
 int j; osc=dither(); // Oscillator amplitude
 for(j=1;j<=3;j++) {
 modop = pwm_mod_sign*Kmod*(vmpwmk[j]- osc);
 if(modop >= Vz+Vd) modop=Vz+Vd; else
 if(modop <= -(Vz+Vd)) modop = -(Vz+Vd);
 pwmtrkm1[j]=pwmtrk[j]; pwmtrk[j]=modop;}
```

≈ *dither* ( ) ≈ 5A

```
// PWM carrier waveform generation
double tdel, delta_t;
long stepk;
double dither(void) {
 double pslp, period, temp;
  period=1/Fd;
  pslp=4*Ad*Fd; // slope of positive going ramp
  temp=fmod(fabs(stepk*delta_t+tdel),period);
  if(temp<period/2)
  return (pslp*temp-Ad); // Pos. going ramp.
  else
  return (Ad-(temp-period/2)*pslp);}
  //Neg. going ramp.
```

Fig. 4-5. Simulation of PWM

The o/p status of the comparator modulator is examined by comparing the trapped value *cpwmtrk***[j]** at the beginning of the time step t_{k-1} with the new o/p *pwmtrk***[j]** at t_k for each phase **j** and signalling any change via the *pwm_test_flag* in the function call *test_pwm_xover* **( )** detailed in Figure 4-6**.** If a crossover event occurs during simulation then the transition interval $\Delta t_X = (t_X - t_{k-1})$, denoted by *min-time***,** is determined by the regula falsi method in (LXVII) in the previous chapter.

```
≈ test_pwm_xover (&pwm_test_flag) ≈
// PWM Pulse Edge Transition Time Detection
void test_pwm_xover(int *flag)
{
 int i, j, ref_sign, act_sign, sigfl;
 double min_time, tol;
 *flag=0; // Reset pwm_sw_flag
 tol=0.001; // Tolerance limit on the time resolution
 for(j=1;j<=3;j++) { // Examine all 3 Φ for PWM X-over
 sigfl=0;
 if(cpwmtrk[j]<0.0) ref_sign = -1; else ref_sign=1;
 if(pwmtrk[j]<0.0) act_sign = -1; else act_sign=1;
 if(ref_sign!=act_sign) { // PWM Crossover Check
   min_time=delt*(cosc-cvmpwmk[j])/(vmpwmk[j]
        -cvmpwmk[j]-osc+cosc); // Expression (3.85)
   if(min_time<0.0)
       nrerror("PWM Switch_time calculation error");
   if(min_time<=(1-tol)*delt) { // switch-time ≤ Δt = T
         if(*flag>=1) {
           for(i=1;i<=(*flag);i++)
             if(min_time>=sw_time[i]-tol*delt &&
               min_time<=sw_time[i]+tol*delt) sigfl=1;
       // Switch times are identical - stall flag increase!
        if(sigfl==0) { ++(*flag); phase_flag[*flag]=j;
         sw_time[*flag]=min_time;} // Store switch time
          } else { ++(*flag); phase_flag[*flag]=j;
                sw_time[*flag]=min_time;}
   } // switch-time ≅ Δt = T
 } else ; // No Crossover!
 } // End 3-phase X-over search!
 if(*flag>0) { // Adjust phase switching times
                // in order of increasing magnitude
   for(i=1;i<=(*flag);i++) { min_time=1.0;
         for(j=i;j<=(*flag);j++) {
           if(sw_time[j]<min_time) {
              min_time=sw_time[j]; ref_sign=j;}
         }
if(i!=ref_sign) {// Define swap (a,b,c): c←a, a←b, b←c
  SWAP(sw_time[i],sw_time[ref_sign],min_time);
  SWAP(phase_flag[i],phase_flag[ref_sign],act_sign);}
   }
 }
 return;}
```

Fig. 4-6. Search for PWM X-over

A check is made to see if this value occurs within the imposed tolerance limit (*tol*delt*) at the end of the time step interval Δt_k, denoted by *delt*, in which it is discarded in the affirmative without test flag registration. If multiple crossover events occur within the simulation interval, corresponding to different phases, then all transition times with the appropriate phase tag number are logged in the respective *sw_time***[j]** and *test_flag***[j]** arrays along with the signaled transition count via the PWM test flag. A check is also performed for identical multiple switch transition times without an increase in test flag count. The test routine is completed by arranging the multiple switch times, with corresponding phase listing, in increasing order of magnitude for subsequent detailed PWM simulation in the function call *inter_pwm_simulation* **( )**. Accurate internal simulation of a modulated pulse transition, indigenous to the time step, commences with restoration of the captured global variables preceding the time step and temporary storage of the original step size (*zeit*) and time delay (*t_del*) settings, relevant to the *dither* **( )** signal source, for later retrieval. The new time step (*delt*) is initially set to the smallest switching interval Δt^1_X (*sw_time***[1]**) for discretization of all first order linear subsystems using the call function *setup_fo_filt* **( )** as per the C-code module in Figure 4-7.

```
      ≈ inter_pwm_simulation (&pwm_sw_flag) ≈
// Simulate 3Φ- PWM with accurate transition times tX
void inter_pwm_simulation(int *flag)
{
    int i,ref_sign,act_sign;
    double zeit, t_del, tol=0.001;
    zeit=delt; t_del=tdel; // Retain original time step info.
              restore_filt_out();
              for(i=1;i<=(*flag);i++) {
  delt=sw_time[i]; // Adjust delt= (tX -tk) to X-over time tX
                 if(i>1) delt -= sw_time[i-1];
                 tdel=t_del-zeit+sw_time[i];
                 setup_fo_filt();
                  run_to_pwmsw();
                if(pwmtrk[phase_flag[i]]<0.0) act_sign = -1;
                else act_sign=1;
                if(cpwmtrk[phase_flag[i]]<0.0) ref_sign = -1;
               else ref_sign=1;
    if(ref_sign==act_sign) pwmtrk[phase_flag[i]] *= -1.0;
                  // Force PWM X-over
                 run_post_pwmsw();
              }   // adjust Δt to complete interval (tk - tX)
             delt=zeit-sw_time[*flag];
             tdel=t_del; // Restore original timing to Vtri (t)
             setup_fo_filt();
             run_to_pwmsw(); run_post_pwmsw();
             delt=zeit; // Restore original time step
             setup_fo_filt();
             return;
}
```

Fig. 4-7. PWM X-Over Simulation

The necessary delay offset t_d (*tdel*) is determined by back tracking (Δt_k-Δt^1_X) from t_k for proper time registration in the execution of the carrier function *dither* **( )** and rerun of the call sequence *run_to_pwmsw* **( )** followed by the block function call *run_post_pwmsw* **( )**. The post PWM

simulation call list contains the additional BLMD model basedrive switching features as an embedded layer in the nested *base_drive* **( )** and associated switch event signalling *test_drk_xover* **( )** program routines. The complete BLMD model program is subsequently exercised for othermultiple switch time intervals $\Delta t^{i>1}{}_x$ with updated linear system discretization and adjusted delay offset**.** Termination of the remainder of the original time step simulation is accomplished by setting the integration interval *delt* equal to the time step residue (Δt_k-$\Delta t^{max}{}_X$) followed by the call sequences *run_to_pwmsw***( )** and *run_post_pwmsw* **( )** and exiting to the main program with a reinstatement of original time settings

Numerical BLMD model simulation proceeds to the next program step in the flowchart cycle shown in Figure 3, by processing the call sequence *run_post_pwmsw* **( )**, with the execution of the switch event routine *base_drive* **( )** associated with the basedrive turn-on/off as a consequence of the PWM process. The relevant global variables and arrays associated with this call sequence run are trapped by the command *capture-drk_out* **( )** as a precursor to basedrive simulation. The 'lockout' circuit routine, illustrated in Figure 4-8, consists of the integrating capacitor action when the PWM comparator o/p $v^k{}_{sm}[j]$ exceeds $v^{k-1}{}_{lj}[j]$ and charge dumping when $v^k{}_{sm}[j] < v^{k-1}{}_{lj}[j]$ as shown in Figure 3 for the basedrive **BDJ** with a similar microprogram description for complementary basedrive $\overline{\text{BDJ}}$ operation. The exponential trigger voltage growth on the timing capacitor, due to the inherent RC circuit delay in (LIV) in the previous chapter, along with the basedrive voltage threshold V_{th} (≈0.0) setting determines the inverter turn-on time. The charge dump action by the shunt diode across the delay timing resistor is virtually instantaneous when the switched comparator PWM output $v^k{}_{sm}[j]=K_{mod}\cdot(v^k{}_c[j]-v^k{}_{tri})$ is hard limited to $-v_z$.

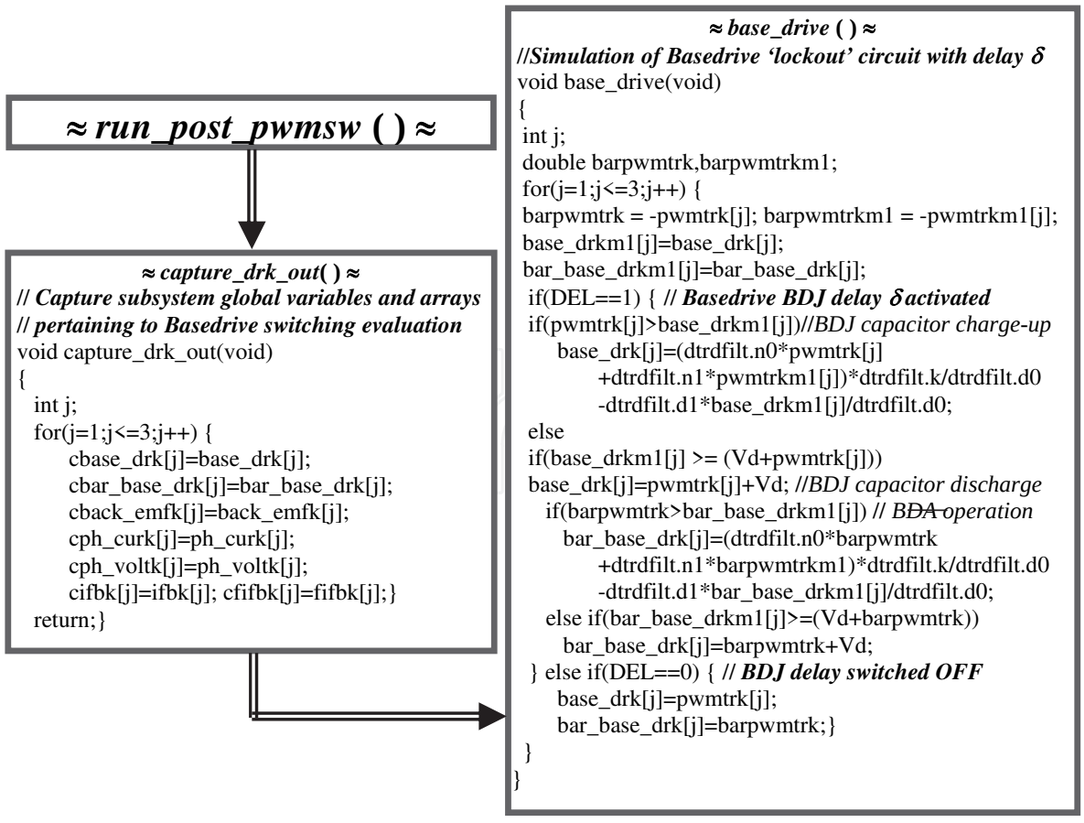


Fig. 4-8. Basedrive simulation

This effect results in swift basedrive turn-off with zero delay when referenced to the trailing edge of the PWM o/p. However the capacitor discharge can be gradual, when the PWM o/p is soft switched ($v_z > v^k_{sm}[j] > -v_z$), due to the limited magnitude of the product combination of modulator gain K_{osc} (~68) and error response $v^k_c[j]$ of the current loop controller which is implicitly dependent on the filtered current feedback response $i^k_f[j]$ for fixed current demand $i^k_d[j]$. The gradual reduction in capacitor voltage protracts the basedrive switch-off time, when referenced to the initial point of the logic "1-to-0" transition, associated with the PWM trailing edge. This delay has to be accounted for in an accurate inverter software model description with a search of the basedrive turn-off in addition to the turn-on times associated with exponential voltage growth.

The BLMD program test function *test_drk_xover* **(&***drk_test_flag***)**, which is shown in Figure 4-10 and is very similar to *test_pwm_xover* **( )** in code content, checks for basedrive on/off firing signal occurrence within a simulation time step interval. This search is complemented with the evaluation of associated multiple phase activation times Δt^i_x for both normal **BDJ** and complementary $\overline{\text{BDJ}}$ inverter drive modes of operation. These inverter trigger instants t^i_x are determined by piecewise linear approximation using (LXXVI) in the previous chapter, ranked in increasing order of magnitude and phase tagged via the global storage arrays *drk_sw_time***[j]** and *drk_phase_flag***[j]** for subsequent use in detailed basedrive simulation. Accurate simulation of the basedrive trigger timing signals for subsequent inverter operation is achieved using the software routine call *inter_base_drk_loop_sim* **(&***drk_test_flag***)** which is shown in Figure 4-11 and has similar execution features to *inter_pwm_simulation* **( )**.

The function call begins with the reinstatement of the global arrays at the beginning (t_{k-1}) of the time step using *restore_drk_out* **( ),** illustrated in Figure 4-9, and temporary storage (*zeit*) of the original step size Δt. The routine proceeds with linear system discretization appropriate to and with execution of the *base_drive* **( )** function and the subsequent call sequence **run_post_drksw ( )**, listed in Figure 4-12, for progressive substitution of multiple differential switch times as the temporary variable *delt*.

This simulation call is completed with restoration of the original time step size followed by first order system discretization with a return to the main BLMD program to begin the new time step $t_k \rightarrow t_{k+1}$. The function call group **run_post_drksw ( ),** summoned during main program execution in the flowchart of Figure 3, processes the following sequence of modular software activities illustrated in Figures 4-12 and 4-13 pertaining to BLMD system electrodynamic operation with inverter interaction.

pwm_inv **( )** generates the 3Φ inverter output HT binary voltage *ph_htk***[j]** $\equiv v^k_g[j]$ in response to the PWM basedrive gating signals v_{bj} & $\overline{v}_{bj}$ shown in Figure 1. The magnitude of the simulated complementary trigger signals v^k_{lj} & $\overline{v}^k_{lj}$ in relation to the basedrive **BDJ** threshold voltage V_{th} establish the conduction states $S_J(k)$ for $k \in \{0,1,2\}$ as per (LV) in the previous chapter, by means of the tristate switching indicator V_O, of the complementary power transistor pair $\mathbf{T}_{J+}$ and $\mathbf{T}_{J-}$ in each leg **J** of the 3Φ inverter shown in Figure 15 in the previous chapter. The tristate flag condition in conjunction with the sustained stator winding current flow *ph_curk***[j]** $\equiv i^k_s[j]$ through the free wheeling shunt diodes establish the inverter o/p binary voltage as 0 or V_{dc}. The neutral star point voltage *Vng* (v_{sg}) of the stator winding is determined from (LVIII) in the previous chapter for subsequent evaluation of the phase voltages *ph_voltk***[j]** $\equiv v^k_s[j]$ via (LIX) in the previous chapter.

```
≈ test_drk_xover (&drk_sw_flag) ≈                    Figure 4-10
// Basedrive switch transition time t_X
void test_drk_xover(int *flag) {
    int i,j,ref_sign,act_sign,sigfl;
    double min_time,tol;
   *flag=0;
    tol=0.001; // Switching time resolution
    for(j=1;j<=3;j++) {
    if(cbase_drk[j]<0.0) ref_sign = -1; else ref_sign=1;
    if(base_drk[j]<0.0) act_sign = -1; else act_sign=1;
// Check basedrive switching times
    if(ref_sign!=act_sign){
       min_time=cbase_drk[j]*delt/(cbase_drk[j]-base_drk[j]);
       if((min_time < 0.0)|| (min_time > delt))
          nrerror("Base Drive Switch_time error");
       if(min_time<=(1-tol)*delta_t) {// switch time < Δt
         if(*flag>=1) {
            for(i=1;i<=(*flag);i++) // identical switch times
               if(min_time>=drk_sw_time[i]-tol*delta_t &&
               min_time<=drk_sw_time[i]+tol*delta_t) sigfl=1;
            if(sigfl==0) { ++(*flag);
              drk_phase_flag[*flag]=j;
               drk_sw_time[*flag]=min_time;}
         } else { ++(*flag);
            drk_phase_flag[*flag]=j;
            drk_sw_time[*flag]=min_time;}
       }
    } else  {
    if(cbar_base_drk[j]<0.0) ref_sign = -1; else ref_sign=1;
    if(bar_base_drk[j]<0.0) act_sign =  -1; else act_sign=1;
    if(ref_sign= =act_sign); else
// Check complementary switch times
if(ref_sign!=act_sign){ ){
    min_time=cbar_base_drk[j]
                *delt/(cbar_base_drk[j]-bar_base_drk[j]);
       if((min_time < 0.0)|| (min_time > delt))
          nrerror("Base Drive Switch_time error");
       if(min_time<=(1-tol)*delta_t) { // switch time < Δt
         if(*flag>=1) {
            for(i=1;i<=(*flag);i++) {// identical switch times
               if(min_time>=drk_sw_time[i]-tol*delta_t &&
               min_time<=drk_sw_time[i]+tol*delta_t) sigfl=1;
            if(sigfl==0) { ++(*flag);
               drk_phase_flag[*flag]=j;
              drk_sw_time[*flag]=min_time;}
         } else { ++(*flag);
            drk_phase_flag[*flag]=j;
            drk_sw_time[*flag]=min_time;}
       }}
  }}
```

```
≈ test_drk_xover (&drk_sw_flag) ≈          Contd
  if(*flag>0) {// Adjust switching times in increasing order
       for(i=1;i<=(*flag);i++) { min_time=1.0;
            for(j=i;j<=(*flag);j++) {
                 if(drk_sw_time[j]<min_time) {
                    min_time=drk_sw_time[j]; ref_sign=j;}
              }
              if(i!=ref_sign) {
SWAP(drk_sw_time[i],drk_sw_time[ref_sign],min_time);
SWAP(drk_phase_flag[i],drk_phase_flag[ref_sign],act_sign);}
    }}
return;}                                          Figure 4-10
```

Fig. 4-10. Search for Basedrive X-Over

```
≈ restore_drk_out ( ) ≈
// Global array restoration for evaluation
// of basedrive switch transition time
void restore_drk_out(void)
{
    int j;
    for(j=1;j<=3;j++
        base_drk[j]=cbase_drk[j];
        bar_base_drk[j]=cbar_base_drk[j];
  back_emfk[j]=cback_emfk[j];
ph_voltk[j]=cph_voltk[j];
        ph_curk[j]=cph_curk[j];
        ifbk[j]=cifbk[j]; fifbk[j]=cfifbk[j];
    }
    return;}
```

Fig. 4-9. Restore basedrive variables

```
≈ inter_base_drk_loop_sim (&drk_sw_flag) ≈
//Simulate basedrive with accurate switching times
void inter_base_drk_loop_sim(int *flag)
{
  int i;
  double zeit;
  zeit=delt; // Retain original time step info.
  restore_drk_out(); // Recover global variables
  for(i=1;i<=(*flag);i++) {
delt=drk_sw_time[i]; // simulate to switching instant t_X
      if(i>1) delt - = drk_sw_time[i-1];
    setup_fo_filt(); base_drive(); run_post_drksw(); }
delt=zeit-drk_sw_time[*flag];
  setup_fo_filt(); //Complete time step interval Sim^n.
  base_drive(); run_post_drksw();
  delt=zeit; // Restore original time step interval Δt
  setup_fo_filt();
  return;}
```

Fig. 4-11. BDJ X-over Simulation

```
≈ run_post_drksw ( ) ≈
≈ pwm_inv ( ) ≈
// Inverter O/P and Motor Phase Voltage Generation
double vng;
int test_flag;
void pwm_inv(void)
{
 int j,vo; // Vo ≡ inverter Tristate switching indicator
 test_flag=0; // Phase winding count with zero current flow
 vng=0.0; // Stator winding neutral voltage w.r.t. ground
// PWM Inverter Conduction State
   for(j=1;j<=3;j++) { flag_cur[j]=0;
       ph_htkm1[j]=ph_htk[j];  ph_voltkm1[j]=ph_voltk[j];
       if(base_drk[j]>Vth && bar_base_drk[j]<Vth) vo=1;
       else
       if(base_drk[j]<Vth && bar_base_drk[j]>Vth) vo = -1;
       else
       if(base_drk[j]<Vth && bar_base_drk[j]<Vth) vo=0;
       else nrerror("NO PWM Inverter Control");
if((vo==1) || (vo==0 && ph_curk[j]<0.0)) ph_htk[j]=Vdc;
else
if((vo == -1) || (vo==0 && ph_curk[j]>0.0)) ph_htk[j]=0.0;
else { test_flag += 1; flag_cur[j]=j;}
   } // Motor Winding Phase Voltage Generation
if((test_flag == 3) || (test_flag == 2))
   for(j=1;j<=3;j++) ph_voltk[j]=back_emfk[j];
if(test_flag == 0) {
   for(j=1;j<=3;j++) vng += ph_htk[j]/3;
   for(j=1;j<=3;j++) ph_voltk[j]=ph_htk[j]-vng;}
if(test_flag==1) {
   for(j=1;j<=3;j++) if(j==flag_cur[j]) test_flag=j;
   ph_htk[test_flag]=back_emfk[test_flag];
   for(j=1;j<=3;j++) vng += ph_htk[j]/2.0;
   ph_htk[test_flag] += vng;
   for(j=1;j<=3;j++) ph_voltk[j]=ph_htk[j]-vng;
   test_flag=0;}
return;}
```

Fig. 4-12. Post Basedrive Operation

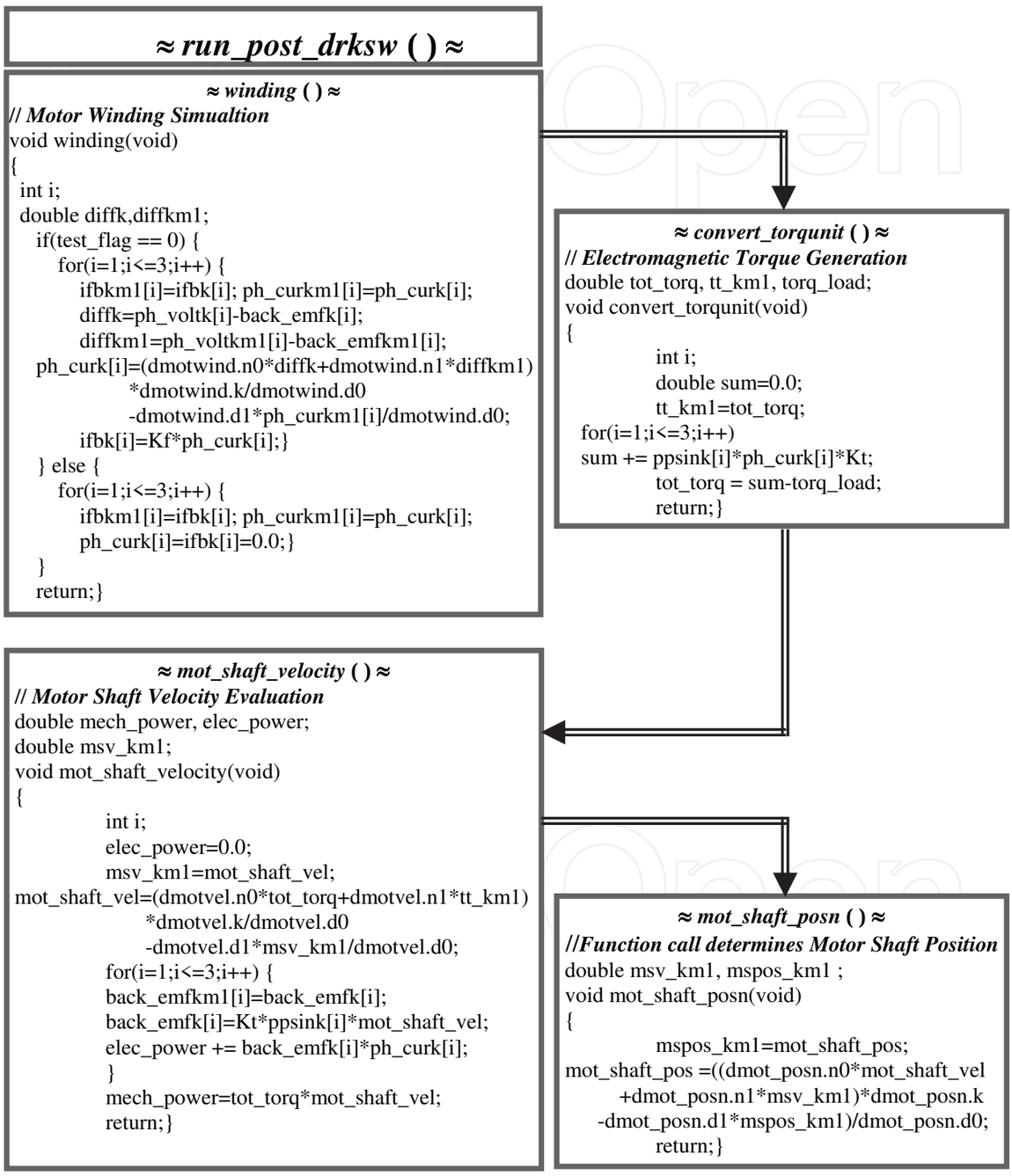


Fig. 4-13. Post Basedrive Call Sequence For Inverter Fed Motor Operation

If one or more phase currents are zero during the condition $V_O = 0$ a *test_flag* ≡ *tf* is incremented by unity for each null occurrence and the relevant phases are tagged using the *test_cur***[j]** array for later identification in the computation of the relevant motor winding phase voltages. A test flag count of 2 or 3 indicates that the phase voltages are identical to the back *EMF* voltages *back_emfk***[j]** ≡ v^k_e[j] from the previous simulation step using expressions (XXIII) and (LVII) in the previous chapter. If the test flag (u) value is unity the associated phase current *ph_curk*[*test_flag*] is zero and the following relations result

$$\begin{aligned} v_{jg} &= v_{js} + v_{sg} = r_s i_{js} + L_s \frac{di_{js}}{dt} + v_{ej} + v_{sg}\,;\forall j \neq tf \equiv u \\ v_{us} &= v_{eu}\,; \quad \because i_{us} = 0 \\ v_{sg} &= \tfrac{1}{2}[\sum_{j\neq u} v_{jg} + v_{us}]\,;\because \sum_{j\neq u} i_{js} = 0 \;\&\; \sum_{j} v_{ej} = 0 \end{aligned} \tag{I}$$

which enable the phase voltages to be calculated from the 3Φ inverter o/p.

winding **( )** determines the stator winding current *ph_curk* **[j]** from a knowledge of the phase voltages and motor parameters {r_s, L_s} encoded in the discrete filter representation of the winding electrical behaviour.

convert_torqunit **( )** computes the developed electromagnetic torque, manifested in the winding current flow, via the 3Φ commutation vector *psink***[j]** and motor torque constant $\mathbf{K}_t$ for dynamic operation.

mot_shaft_vel **( )** evaluates the rotor shaft velocity *mot_shaft_vel* ≡ω^k_m from the net torque *tot_torq* ≡Γ^k_e available, with the effects of load torque retardation *torq_load* ≡Γ^k_l considered, using a discrete filter representation of the rotor dynamics as per (IL) in the previous chapter with parameters {J_m, J_l, B_m}. The back emf can be determined via the motor voltage constant **Ke**, along with the mechanical (*mech_power*) and electrical (*elec_power*) power delivery, once the shaft velocity is known.

mot_shaft_pos **( )** evaluates the motor shaft position *mot_shaft_pos* from the rotor velocity.

3. BLMD model simulation with restraining shaft load torque

The effect of a fixed applied shaft load Γ_l on BLMD model behaviour can be monitored via its simulation characteristics, in torque control mode with and without impedance angle compensation considered, for a range of torque demand step input stimuli capable of matching the posed restraining torque. A suitable choice for the target load magnitude is based on the manufacturers continuous rated stall torque of 5Nm for the particular motor type specified in Table I of the previous chapter. The set of motor shaft velocity step response characteristics $\{\omega_r \,|\, 4\text{v} \le \Gamma_d \le 9\text{v}\,,\, \Delta\Gamma_d = 1\text{v}\}$ for the indicated torque demand Γ_d range is displayed in Figures 6 and 7 without impedance angle compensation.

The response time T_{res} required for the motor drive to reach the maximum shaft velocity of $\omega_{r\max} = 419\ \text{rads.sec}^{-1}$ in each case, corresponding to the rated output of 4000 rpm in Table I of the previous chapter, improves with increased motor shaft acceleration via the "dynamic factor" $[(\Gamma_e - \Gamma_l)/J_m]$ as a consequence of its dependency on torque demand input. The corresponding generated torque step response family of characteristics

$\{\Gamma_e \mid 4\text{v} \le \Gamma_d \le 9\text{v} \;\&\; \Delta\Gamma_d = 1\text{v}\}$ for the specified variation in torque demand inputs are shown in Figures 8 and 9.

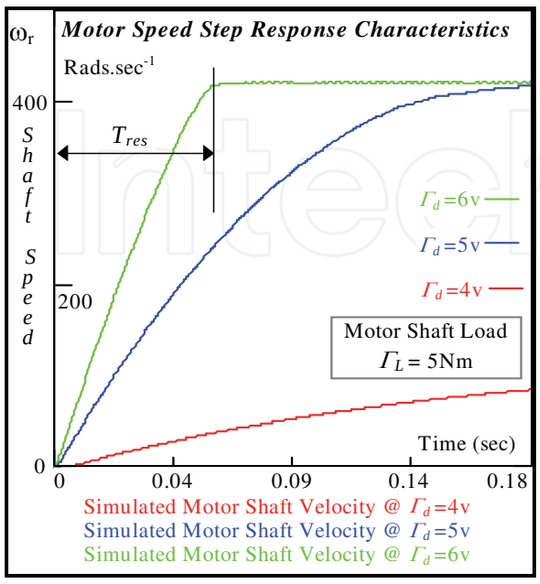


Fig. 6. Shaft Velocity Step Response

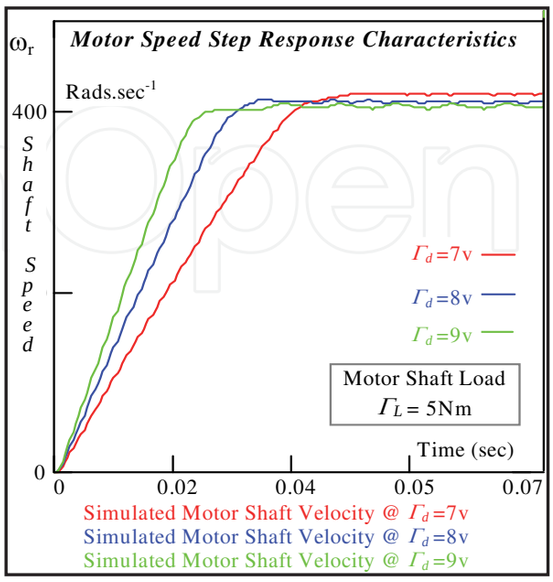


Fig. 7. Shaft Velocity Step Response

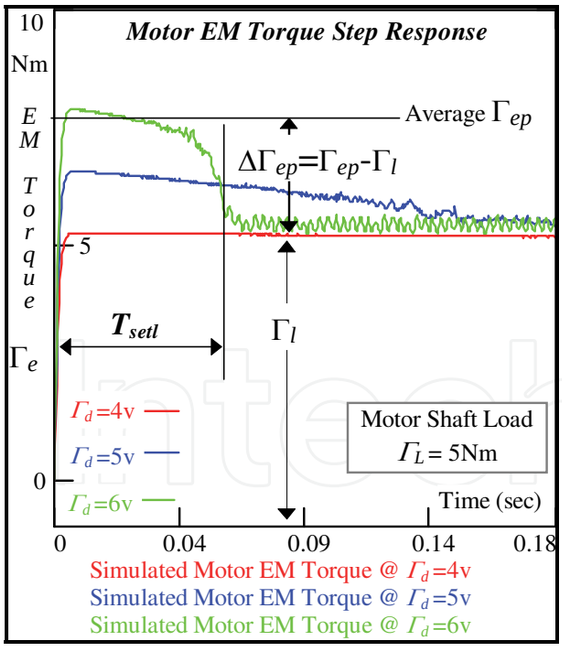


Fig. 8. Generated Torque Response

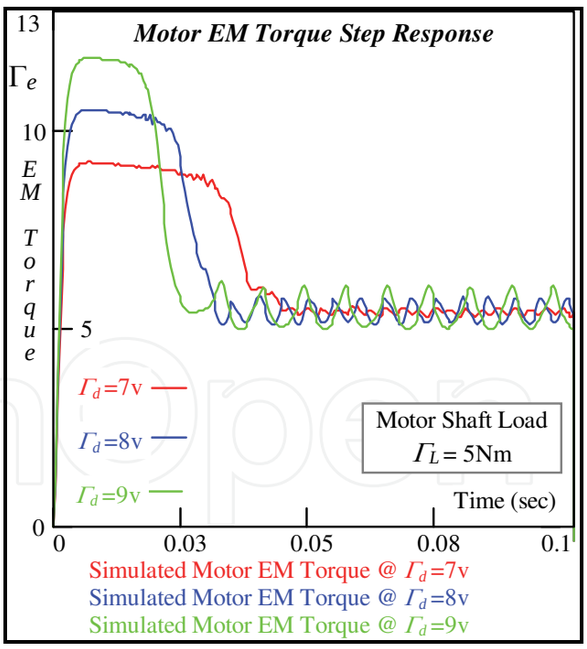


Fig. 9. Generated Torque Response

All the torque response characteristics, with the exception of that at Γ_d = 4v, exhibit overshoot before settling to the required value of ~5.3Nm to overhaul the fixed restraining load torque (5Nm) and frictional effects. The degree of overshoot increases in proportion to the torque demand i/p, as exhibited in Figure 10 for the average peak EM torque Γ_{ep}

responsible for overshoot $\Delta\Gamma_{ep}$, accompanied by a corresponding reduction in settling time as shown in Figure 11.

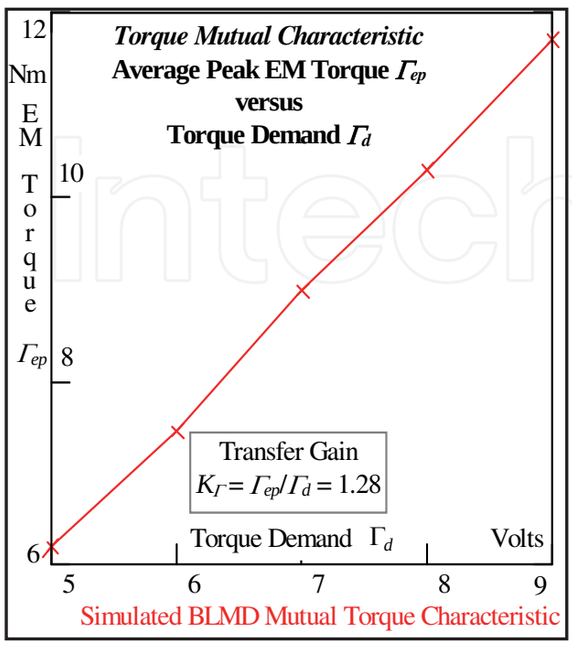


Fig. 10. Mutual Torque Characteristic

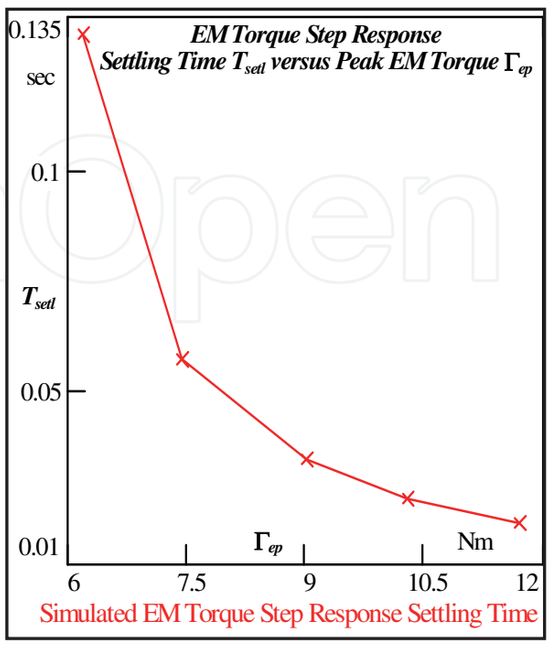


Fig. 11. Torque Settling Time

3.1 Theoretical consideration of motor accelerative dynamical performance

The reduction in settling time is paralleled by the shaft velocity response time improvement in reaching rated motor speed. It is evident from inspection of the velocity and torque simulation traces that a direct correlation exists between the EM torque settling time and motor shaft velocity response time as indicated in Table I.

Torque Load Γ_l =5Nm		"Inertial" Time Constant $\tau_m = J_m / B_m$ =0.318			Tran. Gain (Fig.10) K_Γ =1.28	
Torq-Dem Γ_d (Volts)	Av. Peak EM Torq Γ_{ep} (Nm)	Torque Overshoot $\Gamma_{ep} - \Gamma_l$ (Nm)	Torq. Settling Time T_{setl} (sec) Figs. 8/9	Shaft Velocity Rise Time T_{res} (sec) Figs. 6/7	Theoretical Rise Time T_r (sec) Eqn. (IV)	Rise Time T_α (sec) via Dyn-Fac. Eqn (VI)
5	6.2	1.2	~0.13	~0.13	0.131	0.107
6	7.45	2.45	~0.06	~0.06	0.057	0.0524
7	8.98	3.98	~0.04	~0.037	0.034	0.0323
8	10.29	5.29	~0.03	~0.027	0.025	0.0243
9	11.634	6.634	~0.024	~0.022	0.02	0.02

Table I. Correlation of EM Torque Settling Time with Shaft Velocity Response Time

The shaft velocity step response rise time, as defined in Figure 6, can be obtained directly from the solution of the transfer function (XCIX) from the previous chapter in the time

domain with a step input approximation for the average peak torque overshoot $\Delta\Gamma_{ep} = (\Gamma_{ep} - \Gamma_l)$ in Figure 8 as

$$\omega_r(t) = \frac{\Delta\Gamma_{ep}}{B_m}\left[1 - e^{-t/\tau_m}\right] \quad \text{(II)}$$

with time constant

$$\tau_m = J_m / B_m \quad \text{(III)}$$

The step response time, for the shaft velocity under load conditions to reach maximum speed $\omega_{r\max}$, can be determined from (II) for different torque demand i/p and corresponding peak torque values as per the above Table I with

$$T_r = \tau_m \ln\left[\frac{\Delta\Gamma_{ep}}{\Delta\Gamma_{ep} - B_m\omega_{r\max}}\right] \quad \text{(IV)}$$

The estimated rise times are in excellent agreement with the approximate settling and response times obtained from the BLMD model simulation traces. An alternative crude estimate of the response time can be obtained from the motor "dynamic factor"

$$\alpha = \frac{d\omega_r}{dt} = \frac{(\Gamma_{ep} - \Gamma_l)}{J_m} \quad \text{(V)}$$

for average peak torque endurance as the acceleration time

$$T_\alpha = \frac{\omega_{r\max}}{\alpha} \quad \text{(VI)}$$

from standstill to maximum speed assuming a shaft velocity linear transient response which is valid for torque demand values in excess of 5 volts.

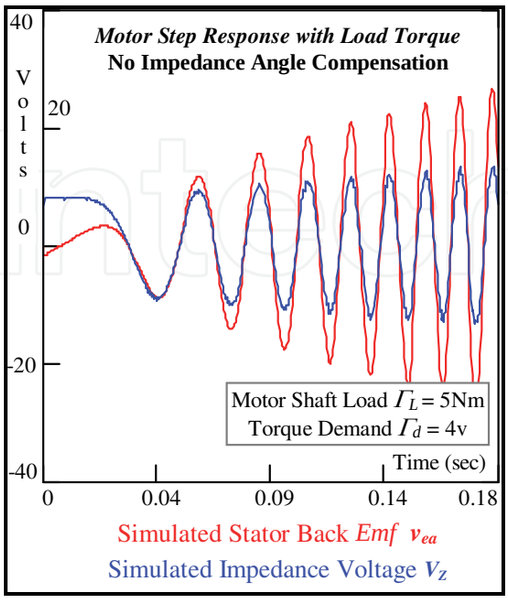


Fig. 12. Motor Winding Voltages

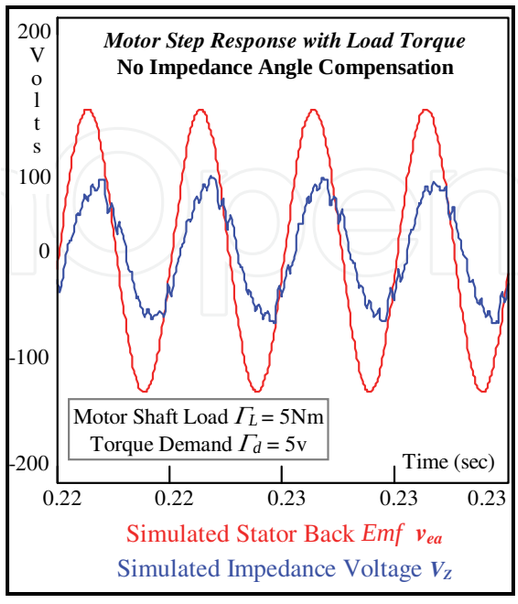


Fig. 13. Motor Winding Voltages

These response estimates, given in above Table I, are in good agreement with those already obtained except for that at Γ_d = 5v where the rise time is longer with exponential speed ramp-up.

3.2 Torque demand BLMD model response - internal node simulation

The simulated back-EMF along with the stator impedance voltage drop are illustrated in Figures 12 and 13 for two relatively close values of torque demand i/p. In the former case the torque demand i/p of 4volts results in sufficient motor torque to meet the imposed shaft load constraint (5Nm) without reaching rated speed and saturation (±10v) of the current compensator o/p trace shown in Figure 14. The corresponding reaction EMF exceeds the winding impedance voltage V_Z and is almost in phase with the stator current, which is proportional to V_Z, at the particular low motor speed reached. The torque demand i/p of 5v in the latter case results in the onset of a clipped current controller o/p in Figure 15 due to saturation (±10) at rated motor speed ω_{rmax}.

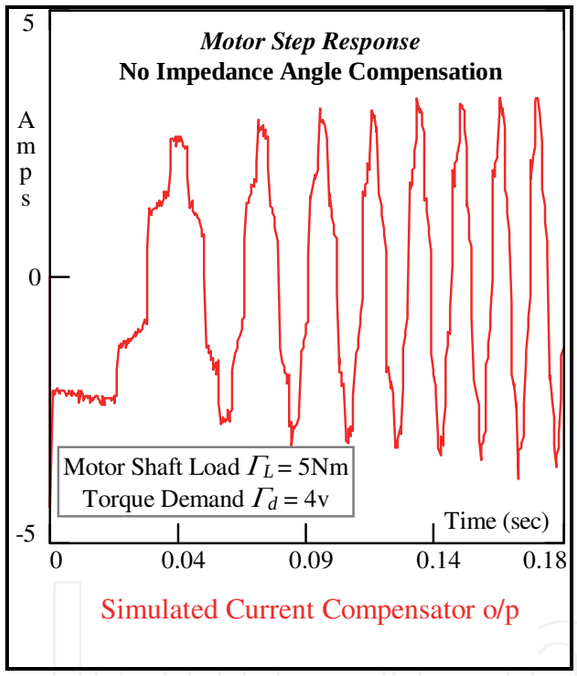


Fig. 14. Current Compensator o/p

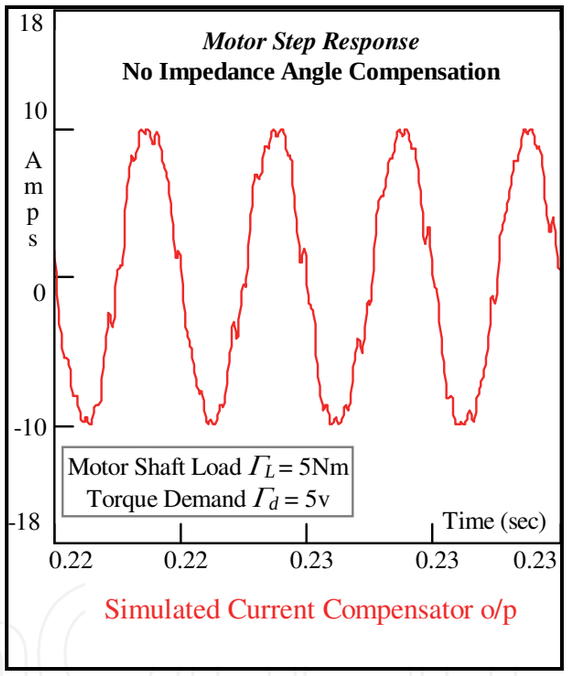


Fig. 15. Current Compensator o/p

The back-EMF generated at this speed greatly exceeds the winding impedance voltage, as in the former case, and leads the stator current necessary to surmount the torque load by the internal power factor (PF) angle (~27°) with a correspondingly low power factor (~0.7).

The stator winding currents corresponding to the inputs $\Gamma_d \in \{5v,9v\}$ are displayed in Figures 16 and 17 respectively which indicate the marked presence of peak clipping in the latter case with loss of spectral purity due to heavy saturation of the current controller o/p for Γ_d >5v.

The simulated motive power characteristic with the steady state threshold value of ~2.3kW necessary to sustain shaft motion, for Γ_d =5v with restraining load torque and friction losses is shown in Figure 18 at base speed $\omega_{rmax} \cong 420$ rad.sec^{-1}.

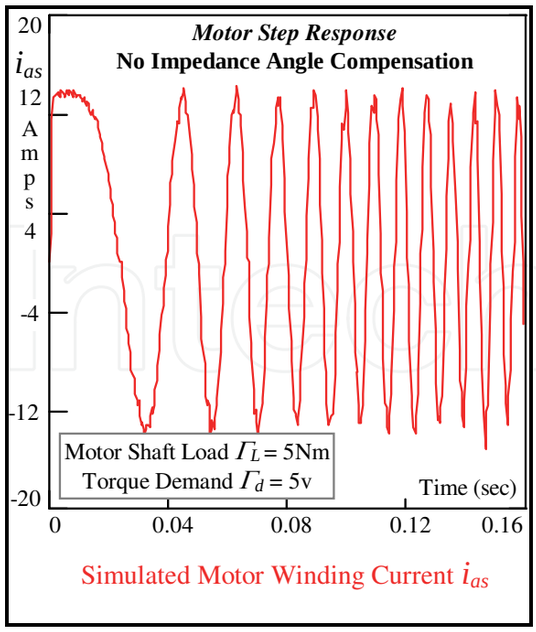


Fig. 16. Stator Winding Current

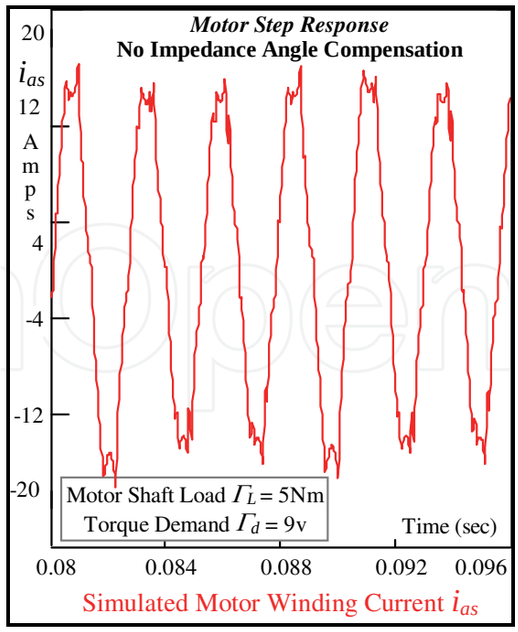


Fig. 17. Stator Winding Current

This can be rationalized from the power budget required to sustain the load torque at rated speed via (LXXXVIII) in the previous chapter as

$$P_l = \Gamma_l \omega_{r\max} = (5)(420) = 2.1\text{kW} \qquad \text{(VII)}$$

The excess coupling field power required to surmount mechanical shaft friction losses is shown simulated in Figure 19 with a steady state estimate of ~200 watts.

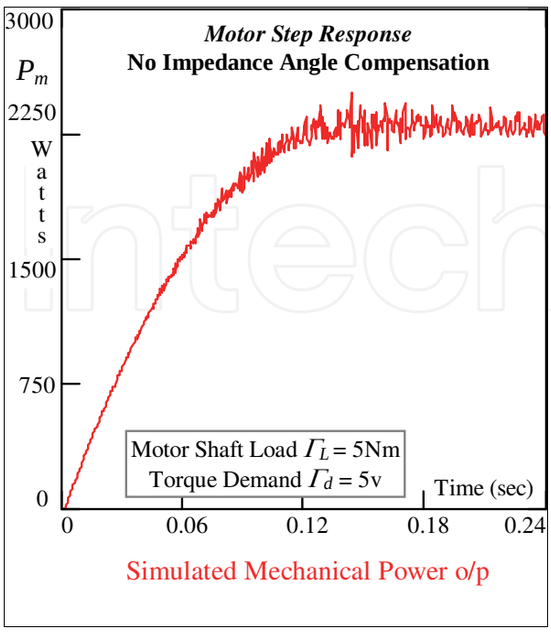


Fig. 18. Mechanical Power Delivery

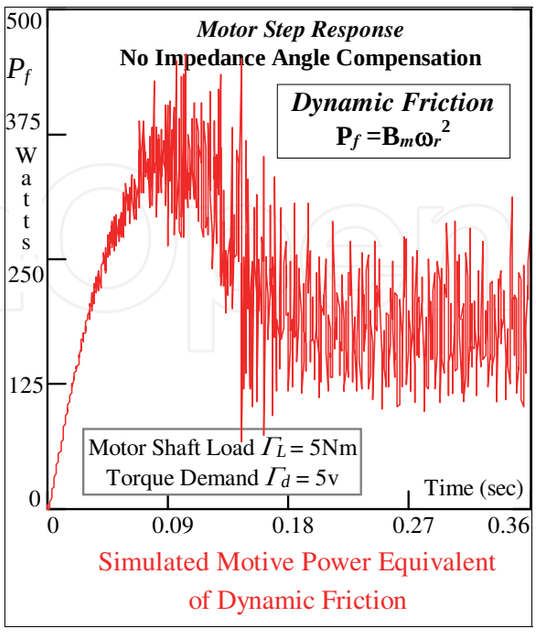


Fig. 19. Dynamic Friction Loss

Stator Winding Phasor RMS Magnitude Estimation as per Figure 44 in previous chapter via BLMD Model Simulation						
Torq_Dem Γ_d Step i/p	Shaft_Vel ω_{rmax} rad.sec^{-1}	Elec_Power P_e volts (XLVII) – Prev. Chap	Back_EMF $\lvert V_{ej} \rvert$ volts	Imped_Vol $\lvert V_Z \rvert$ volts (XC) – Previous Chap	Ph_Cur $\lvert I_{js} \rvert$ amps	
5v	419.2	2301	94.82	44.87	9.09	
6v	420.3	2305	95.24	48	9.7	
7v	418.9	2298	94.78	50.89	10.32	
8v	410.3	2251	92.05	52.42	10.84	
9v	405.5	2224	91.5	53.66	11.23	
Derived Phase Quantities as per Figure 42 in previous chapter						
Torq_Dem Γ_d volts	Int. PF Ang φ_I (XCII) – Prev. Chap.	φ_I Estimate via Figure 13	Ph_Vol V_{js} (XCIII) – Prev. Chap	Imp_Ang φ_Z (LXXXIV) – Prev. Chap.	Load Ang β_T (XCV) – Prev. Chap	PF Ang φ $\varphi_I + \beta_T$
5	27.13°	27.13°	126.44v	81.26°	15.52°	42.65°
6	33.7°	32.16°	131.75v	81.28°	15.6°	49.3°
7	38.43°	38.74°	136.56v	81.26°	14.68°	53.11°
8	41.24°	42.58°	136.5v	81.08°	13.83°	55.07°
9	43.8°	45.12°	138.1v	80.97°	13.58°	57.38°

Table II. Phase Angle Evaluation for BLMD Steady State Operation with Γ_l = 5Nm

The effect of shaft load on the BLMD model simulation characteristics for Γ_d >5v is summarized in above Table II for steady state conditions with the aid of the general phasor diagram in Figure 42 of the previous chapter.

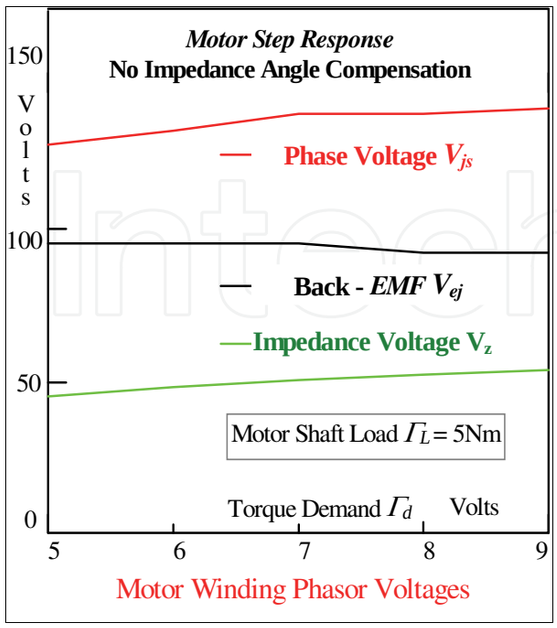


Fig. 20. Motor RMS Phasor Voltages

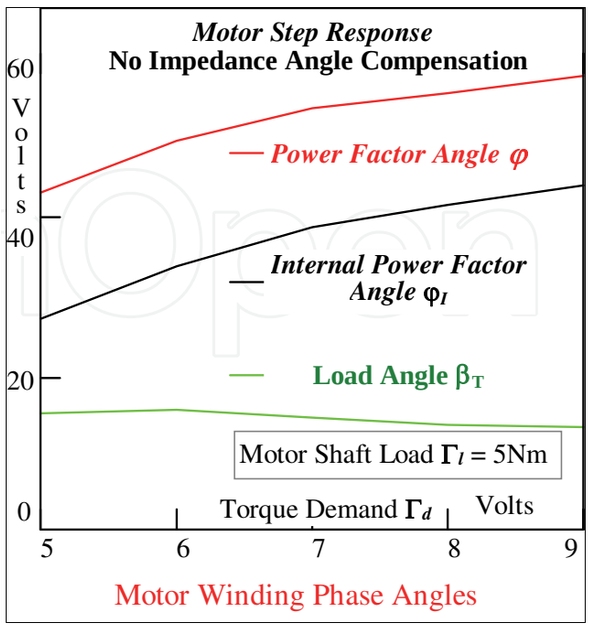


Fig. 21. Stator Winding Phase Angles

It is evident from the table that the back EMF has reached its peak rms value with the onset of maximum shaft velocity, for all values of Γ_d >5V, with

$$V_{ej_{\max}} = \frac{K_e}{\sqrt{2}}\omega_{r_{\max}} = \left(\frac{0.315}{\sqrt{2}}\right)(420) = 93.6\text{V} \tag{VIII}$$

Furthermore the impedance voltage drop V_z in (XC) of the previous chapter is limited to a very small increase with torque demand current I_{dj} listed in Table II and is shown almost stabilized to a constant value in Figure 20. This voltage clamping effect, due to current compensator o/p saturation in response to tracking current feedback, is controlled to achieve the desired rms level of clipped current flow in the stator winding as shown in Figure 17 to satisfy torque load requirements. The rms winding current flow necessary at unity internal power factor to meet steady state toque load and friction demands at ~5.4Nm in Figures 8 and 9 can be determined from (XLV) in the previous chapter as

$$\hat{I}_{js} = \left(\frac{\sqrt{2}}{3}\right)\left(\frac{\Gamma_e}{K_e}\right) = \left(\frac{\sqrt{2}}{3}\right)\left(\frac{5.42}{0.315}\right) = 8.11\text{ Amps} \tag{IX}$$

This is almost identical to the rms values obtained from BLMD model simulations in Table II, which are consistent with increased torque current demand, when internal power factor self adjustments are accounted for as in

$$\hat{I}_{js} = I_{js}\cos\varphi_I \approx 8.1\text{ Amps} \tag{X}$$

The internal power factor angles, listed in Table II and displayed in Figure 21, are deduced for Γ_d >5v from the mechanical power transfer by substituting the rms quantities obtained from back EMF and winding current simulations in expression (XCII) of the previous chapter. These angles, which increase with torque demand i/p, can be alternatively calculated from the simulated winding current response using (X) with knowledge of $\hat{I}_{js}$.

The tabulated angle estimates obtained statistically as the phase lag between the current and back EMF waveforms in Figure 13, for example, are in close agreement with those from (XCII) of the previous chapter. The motor winding impedance angle φ_z, which is fixed at rated machine speed ω_{rmax}, is determined from (LXXXIV) as ~81.2° in Table II.

The rms winding voltage V_{js} is obtained in its pure spectral form, instead of the PWM version furnished by the current controlled inverter, upon application of (XCIII) to the known rms phasor quantities given in Table II for different values of Γ_d >5V.

Knowledge of the relevant phasor magnitudes with corresponding phase angles enable the load angle β_T to be determined from (XCV) of the previous chapter for given shaft load conditions. This is approximately fixed, at ~15° as indicated in Table II with about 2° variation, over the torque demand i/p range as shown in Figure 21. The resulting power factor angle φ listed in Table II increases with φ_I, for fixed load angle over the torque demand i/p range as shown, in a way that is commensurate in (X) with motor current requirements towards sustaining shaft load torque with a decreasing power factor as illustrated in Figure 22.

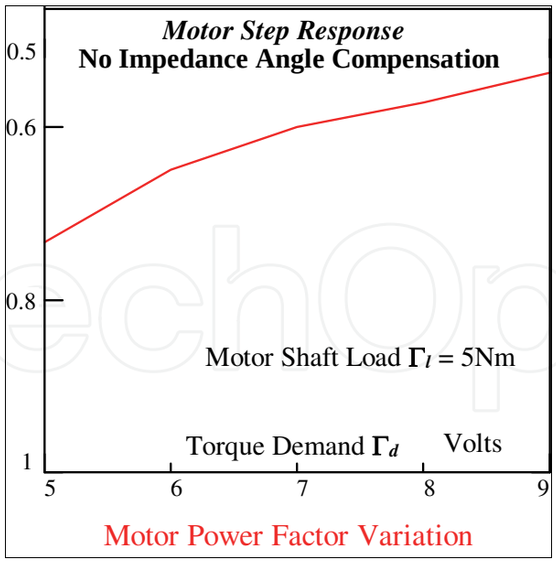


Fig. 22. Power Factor Variation

3.3 BLMD model simulation with novel impedance angle compensation

The effect of motor impedance angle compensation (MIAC), manifested as commutation phase lead angle incorporated into the BLMD model in (XCVIII) of the last chapter as $p\theta_r - 2(j-1)\frac{\pi}{3} \rightarrow (p\theta_r + \varphi_z) - 2(j-1)\frac{\pi}{3}$ on the motor step response velocity and torque characteristics is illustrated in Figures 23 and 24 for the torque command i/p range $4\text{v} \le \Gamma_d \le 9\text{V}$ at step size intervals of $\Delta\Gamma_d = 1\,\text{volt}$.

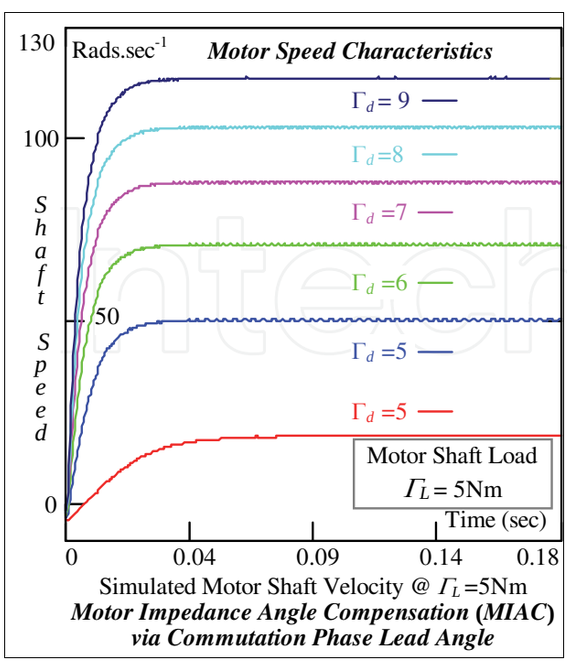


Fig. 23. Shaft Velocity with MIAC

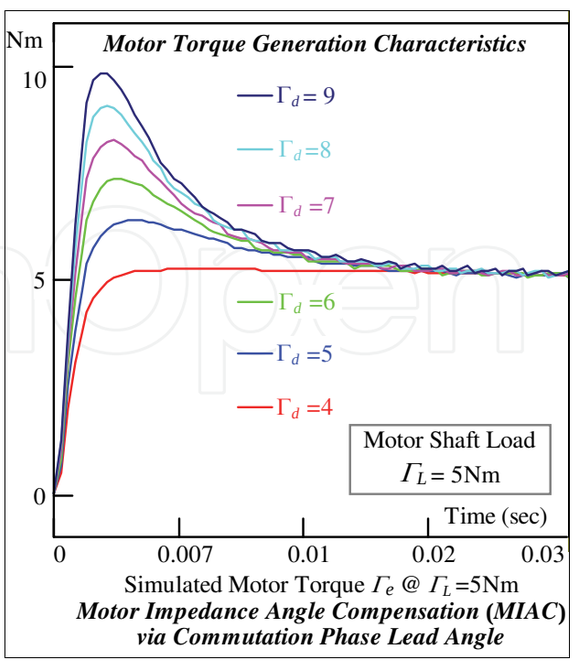


Fig. 24. Torque Response with MIAC

The variation of peak torque overshoot with i/p demand, displayed as the mutual characteristic in Figure 25, is linear with a transfer gain that is lower than that without MIAC in Figure 10. Consequently the maximum peak torque delivery, for a given i/p demand to sustain shaft load requirements, is lower in amplitude and of shorter overshoot pulse duration as seen in Figure 24 when compared with that without MIAC in Figures 8 and 9. Furthermore the persistence of torque overshoot is lower with a much reduced settling time (<0.015 sec), in reaching steady state sustained load conditions in all cases albeit at lower acceleration and much smaller drive speeds, thereby exerting less mechanical stress on the drive shaft components and minimizing shaft flexure in EV propulsion applications.

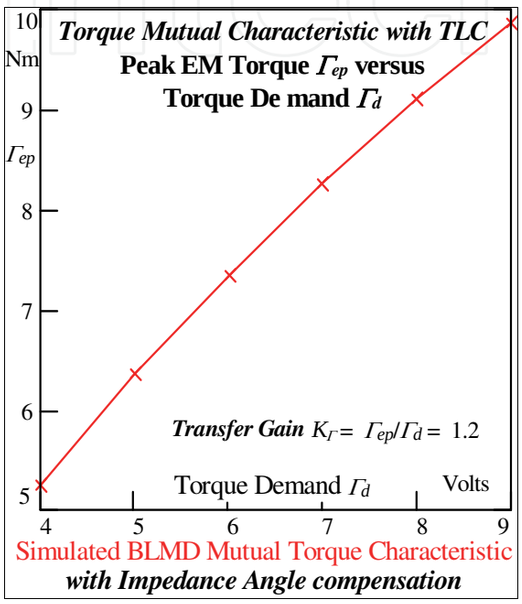


Fig. 25. Mutual Torque with MIAC

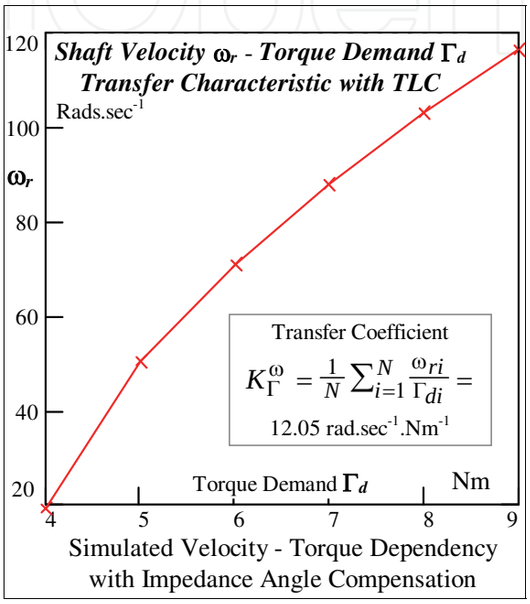


Fig. 26. Torque - Velocity Transfer Curve

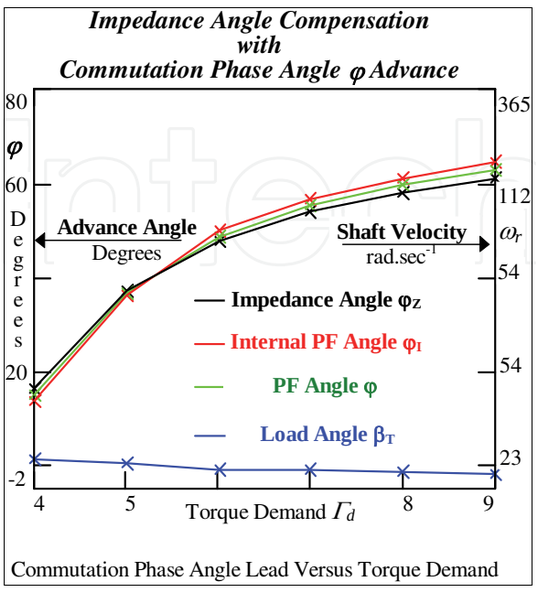


Fig. 27. Impedance Angle Compensation

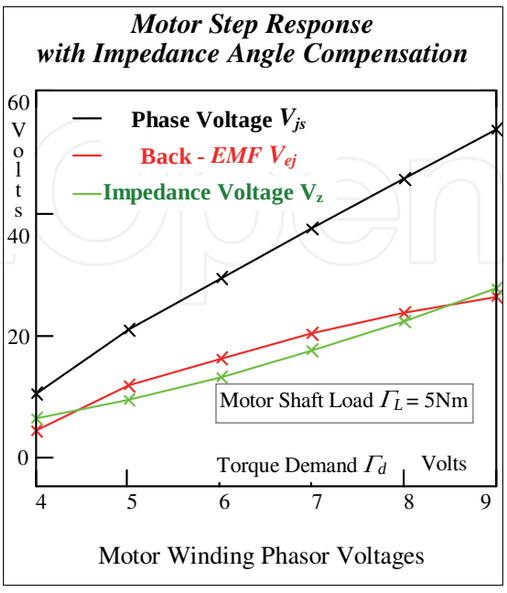


Fig. 28. Phasor Voltages with MIAC

The shaft velocity characteristics also indicate a much lower steady state motor run speed, with MIAC deployed, which never reaches velocity saturation $\omega_{r\max} = 419\,\text{rads.sec}^{-1}$ over the permissible torque demand i/p range of $-10\text{V} \le \Gamma_d \le 10\text{V}$. The relevant command torque to shaft velocity transfer characteristic is approximately linear as shown in Figure 26 which indicates a maximum motor operating speed of $\omega^*_{r\max} \approx 120\,\text{rad.sec}^{-1}$ with $\omega^*_{r\max} < 30\%\omega_{r\max}$ under rated load conditions (5Nm) for a maximum demand i/p of Γ_{dmax} =10V. This speed reduction is singly due to the maintenance of an almost zero load angle β_T shown in Figure 27, between the motor terminal V_{js} and back EMF V_{ej} rms voltage phasors in Figure 45 of the previous chapter, by commutation phase angle advance for optimal torque production as indicated from the BLMD simulation results in Table III.

This phase compensation technique results in back EMF and winding impedance voltage V_z phasors that appear approximately equal in magnitude over the allowable torque demand input range as shown in Figure 28. Furthermore the internal power factor angle φ_I is forced to adopt approximately the same value as the machine impedance angle φ_z as indicated in Table IIII, by the phase advance measure φ_z in the current commutation circuit, with a consequent collinear alignment of phasors V_{ej} and V_z in Figure 45. This collinear arrangement can only be sustained at a particular machine speed that is dependent on the torque demand i/p which determines the subsequent winding current flow and thus the necessary impedance angle for alignment. This reasoning can be deduced as follows by noting that for a given torque load Γ_l the rms winding current flow is linear with torque demand i/p as per Table III and Figure 29.

Stator Winding Phasor RMS Magnitude Estimation as per Figure 45 of the Previous Chapter via BLMD Model Simulation					
Torq_Dem Γ_d Step i/p	Shaft_Vel ω_{rmax} rad.sec^{-1}	Elec_Power P_e (XLVII) in Prev. Chap.	Back_EMF V_{ej} volts	Imped_Vol V_Z volts – (XC) in Prev. Chap.	Ph_Cur I_{js} amps
4v	18.6	94.44	4.17	6.06	7.76
5v	48.95	257.2	11.5	9.23	9.7
6v	70.87	363.67	16.01	13.05	11.71
7v	87.9	452.6	19.95	17.33	13.66
8v	102.9	531.2	23.28	22.18	15.7
9v	116.3	602.2	26.3	27.45	17.74
Derived Phase Quantities as per Figure 42 of the Previous Chapter					
Torq_Dem Γ_d volts	Int. PF Ang φ_I (XCII) in Prev. Chap.	Ph_Vol V_{js} (XCIII) in Prev. Chap.	Imp_Ang φ_Z (LXXXIV) in Prev. Chap.	Load Ang β_T (XCV) in Prev. Chap.	PFAng φ $\varphi_I + \beta_T$
4	13.75°	10.23	16.1°	1.39°	15.14°
5	36.13°	20.73	37.22°	0.51°	36.64°
6	49.71°	29.06	47.72°	-1.0°	48.71°
7	56.38°	37.27	53.76°	-1.22°	55.16°
8	61.02°	45.44	57.95°	-1.5°	59.52°
9	64.52°	53.73	61.01°	-1.79°	62.73°

Table III. Phase Angle Evaluation at Γ_l = 5Nm with Motor Impedance Angle Compensation

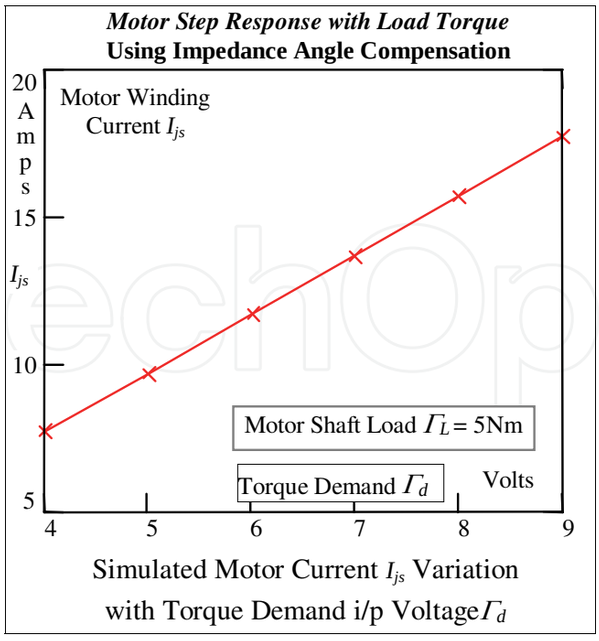


Fig. 29. Motor Current Variation

3.3.1 MIAC substantiation via theoretical analysis and validation

The internal power factor angle φ_I can be determined theoretically for fixed winding current flow corresponding to a given torque demand i/p using (IX) and (X), assuming negligible dynamic friction at the shaft speeds concerned with $\Gamma_l >> \Gamma_f$, as

$$\varphi_I = \cos^{-1}\left\{\left(\frac{\sqrt{2}}{3}\right)\frac{\Gamma_l}{K_t I_{js}}\right\} \tag{XI}$$

The motor terminal voltage i/p V_{js} in (XCIII) from previous chapter can be optimized with respect to the motor impedance angle φ_z, which is unknown, in terms of the rms phasor quantities V_{ej}, V_z and the fixed internal power angle φ_I from (XI) by letting

$$\frac{dV_{js}}{d\varphi_z} = 0 \Rightarrow \sin(\varphi_z - \varphi_I) = 0 \tag{XII}$$

This procedure results in the impedance angle φ_z in terms of the known angle φ_I as

$$\varphi_z = \varphi_I \tag{XIII}$$

with $$max\, V_{js} = \sqrt{V_{ej}^2 + V_z^2 + 2V_{ej}V_z} = (V_{ej} + V_z) \tag{XIV}$$

which is unknown as both V_{ej} and V_z depend on the motor shaft velocity ω_r. The shaft velocity can now be determined from (LXXXIV) from previous chapter using expression (XIII) as

$$\omega_r = \left(\frac{r_s}{pL_s}\right)\tan\varphi_I \tag{XV}$$

Theoretical Estimation of RMS Phasor Magnitudes						
Torq_Dem Γ_d Step i/p	Ph_Cur I_{js} Table III	Int_PF φ_I Eqn. (XI)	Shaft_Vel ω_r Eqn (XV)	Back_EMF V_{ej} Eqn (VIII)	Imp_Vol V_Z Eqn. (XC) in Prev. Chap.	Ph_Vol V_{js} Eqn. (XIV)
4v	7.76 A	15.37°	17.71 rad/s	3.94v	6.04v	9.98v
5v	9.70 A	39.52°	53.17 rad/s	11.84v	9.43v	21.27v
6v	11.71 A	50.28°	77.55 rad/s	17.27v	13.74v	31.01v
7v	13.66 A	56.79°	98.43 rad/s	21.92v	18.71v	40.63v
8v	15.70 A	61.54°	118.87 rad/s	26.48v	24.71v	51.19v
9v	17.74 A	65.05°	138.49 rad/s	30.85v	31.54v	62.39v

Table IV. Motor Impedance Angle Compensation

This value of ω_r can be used to theoretically generate the rms voltage phasors V_{ej}, V_z and V_{js} using expressions (VIII), (XC) and (XCIII) in the previous chapter respectively from a knowledge of the motor winding phasor current I_{js} as per Table IV over the i/p torque demand range range $\Gamma_d \geq 4V$. The quantities obtained from BLMD simulations in Table III compare reasonably well with those derived in Table IV from theoretical considerations which reinforces model validation and confidence. The optimized internal power factor angle, which is almost identical to that in Table III, results in a zero load angle β_T from (XCV) in the previous chapter due to the phasor collinearity and thus improved torque control via the PWM voltage supplied by the current controlled inverter. The power factor angle φ, internal power factor angle φ_I and machine impedance angle φ_z variations with torque demand i/p, which are displayed in Figure 27 using estimates extracted from BLMD model simulation in Table III for $\Gamma_d \geq 4V$, are almost congruent with a mismatched difference manifested as the negligible load angle ($\beta_T \approx 0$).

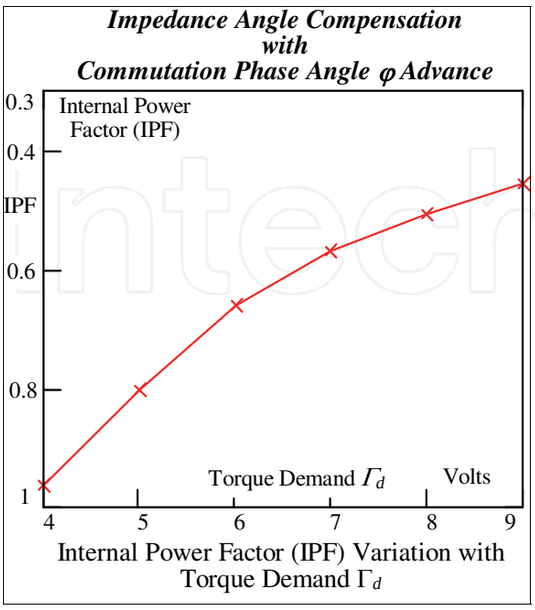


Fig. 30. Motor Power Factor

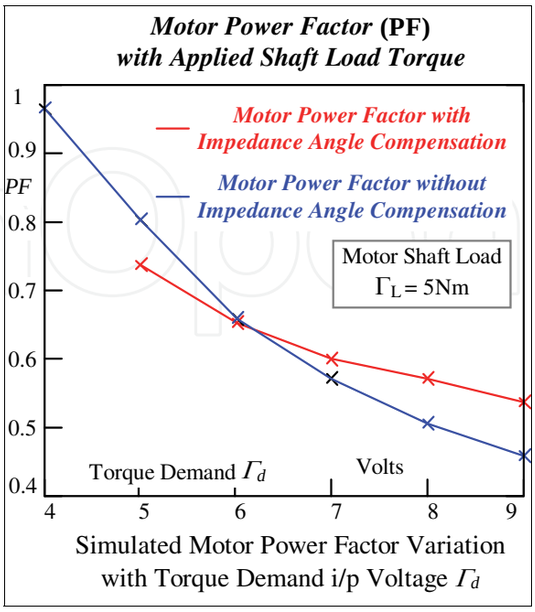


Fig. 31. Power Factor Comparison

The internal power factor $\cos\varphi_I$ shows a gradual deterioration with increasing torque demand i/p in Figure 30 as expected with the accompanying internal power factor angle φ_I adjustment, from the mirrored motor current increase in Figure 29, constrained by a fixed shaft load in (X). Impedance angle compensation results in a improved motor power factor as shown in Figure 31 than that without MIAC over the torque demand i/p range $4\text{V} \le \Gamma_d \le 6\text{V}$ necessary to meet load requirements Γ_l.
Motor speed reduction is also mirrored with a decrease of the shaft velocity step response rise time as shown Figure 32 with maximum values falling below the velocity time response floor of the uncompensated BLMD model. This results in constant motor speed operation, though small by comparison to that without phase angle advance, well below the rated value in torque control mode with smooth torque delivery to satisfy load requirements.

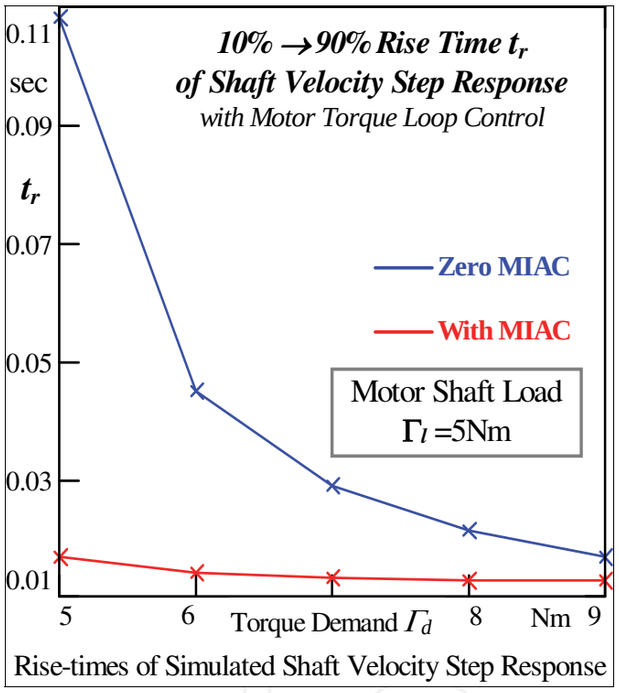


Fig. 32. Shaft Velocity Rise Times

The simulated motor winding impedance and back EMF voltages for mid (5V) and full range (9V) torque demand input values, which result in developed torque capable of surmounting the fixed restraining shaft load (5Nm), are displayed in Figures 33 and 34. Both sets of characteristics exhibit comparable amplitudes appropriate to the level of torque demand i/p, with speed related motor current phase lags φ_I as per Table III, that are much lower than those without MIAC in Figure 13. The impedance and back EMF voltages are interrelated which can be shown as follows by starting with expression (XC) for V_z and using (IX) and (X) giving

$$V_z = |Z||I_{js}| = |Z|\left(\frac{\sqrt{2}}{3}\right)\left(\frac{\Gamma_l}{K_t}\right)\cos\varphi_I \qquad \text{(XVI)}$$

This can be rewritten by using (LXXXIV) in the previous chapter with optimized value of φ_I in (XIII) as

$$V_Z = \left(\frac{\sqrt{2}}{3}\right)\left(\frac{\Gamma_l}{K_t}\right)\left(\frac{r_s^2 + (p\omega_r L_s)^2}{r_s}\right) \tag{XVII}$$

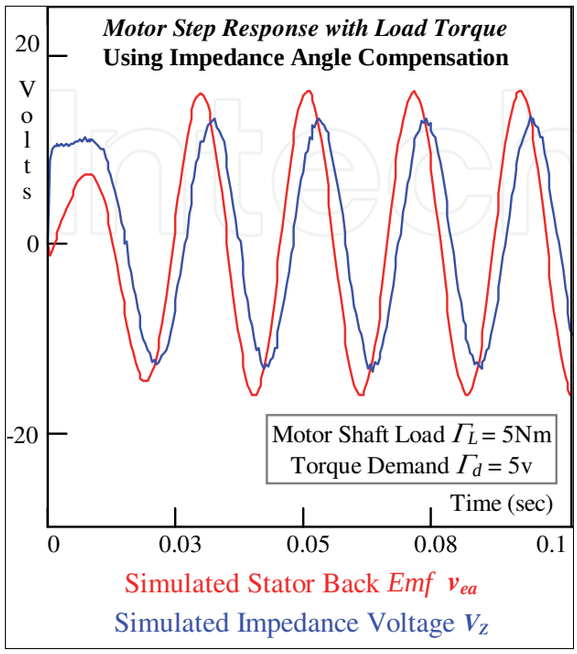


Fig. 33. Motor Winding Voltages

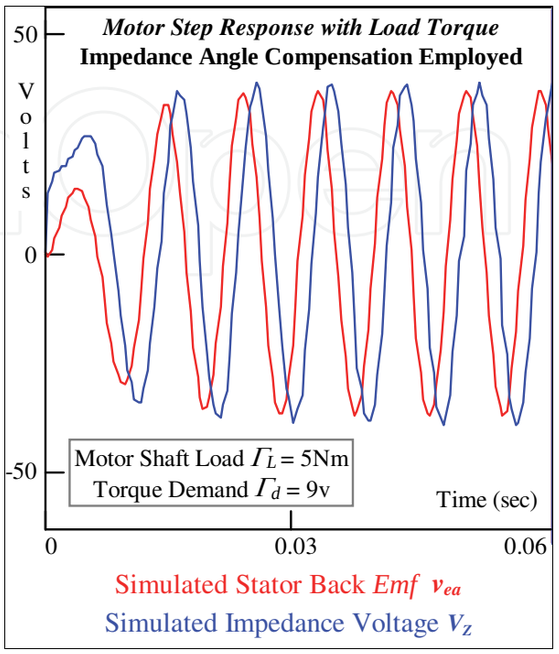


Fig. 34. Motor Winding Voltages

The shaft velocity ω_r, linking the back EMF, can be replaced in (XVII) by using (VIII) yielding

$$V_Z = K_1[1 + K_2 V_{ej}^2] \tag{XVII}$$

where

$$K_1 = r_s\left(\frac{\sqrt{2}}{3}\right)\left(\frac{\Gamma_l}{K_t}\right) = 5.612 \tag{XVIII}$$

and

$$K_2 = \left(\frac{\sqrt{2}\,pL_s}{K_t r_s}\right)^2 = 4.855 \times 10^{-3} \tag{XIX}$$

from substitution of parameters in Table I of the previous chapter and Γ_l = 5Nm. The impedance voltage in (XVII) is expressed as a quadratic equation in terms of the back EMF with points of equality corresponding to

$$V_{ej} \in \left\{6.915\text{v}, 29.79\text{v} \middle| V_Z = V_{ej}\right\} \tag{XX}$$

which are visible in Figure 28 as points of intersection of the two voltage traces.

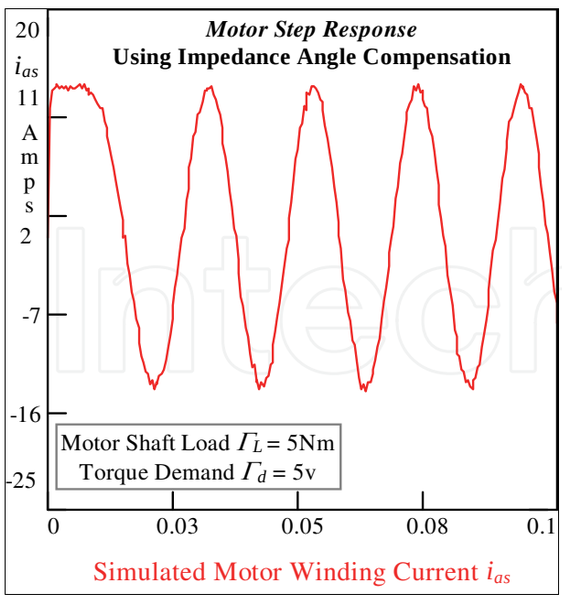


Fig. 35. Stator Winding Current Flow

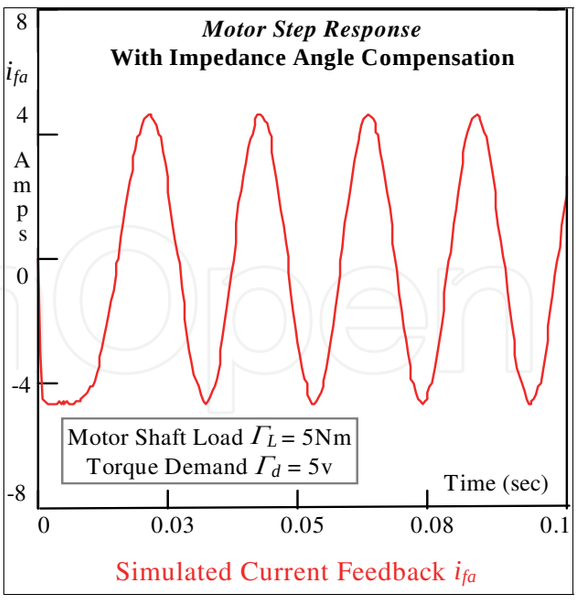


Fig. 36. Motor Current Feedback

These crossover points divide the rms V_z amplitude variation along with V_{ej} in Figure 28 into three distinct regions, over the usable torque demand i/p range as per Table IV, with

$$\begin{aligned} V_z &> V_{ej} \text{ for } V_{ej} < 6.9\text{v} \\ V_z &< V_{ej} \text{ for } 6.9\text{v} < V_{ej} < 29.8\text{v} \\ V_z &> V_{ej} \text{ for } V_{ej} > 29.8\text{v} \end{aligned} \qquad \text{(XXI)}$$

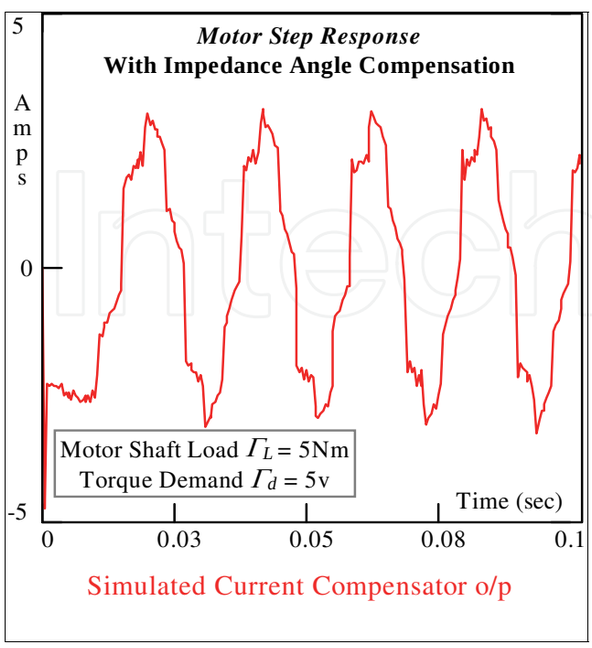


Fig. 37. Current Controller o/p

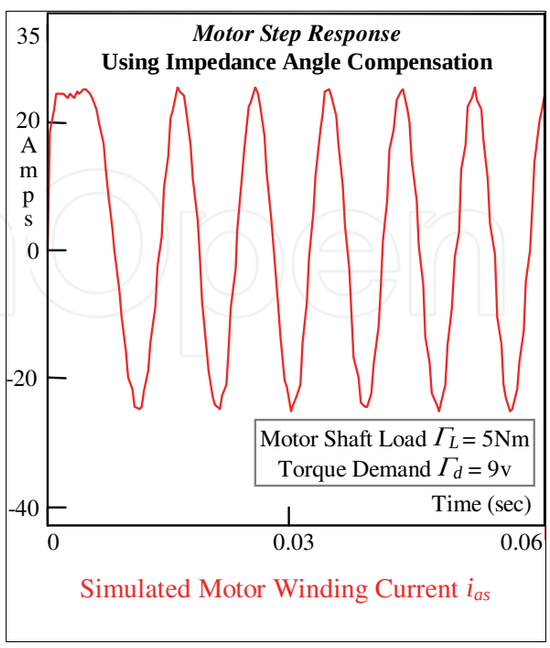


Fig. 38. Stator Winding Current Flow

These regions can also be inferred from the voltage amplitude traces in Figure 33 and 34 where V_{ej} exceeds V_z in the former case with $\Gamma_d = 5\text{v}$ and vice versa for $\Gamma_d = 9\text{v}$ in the latter diagram.

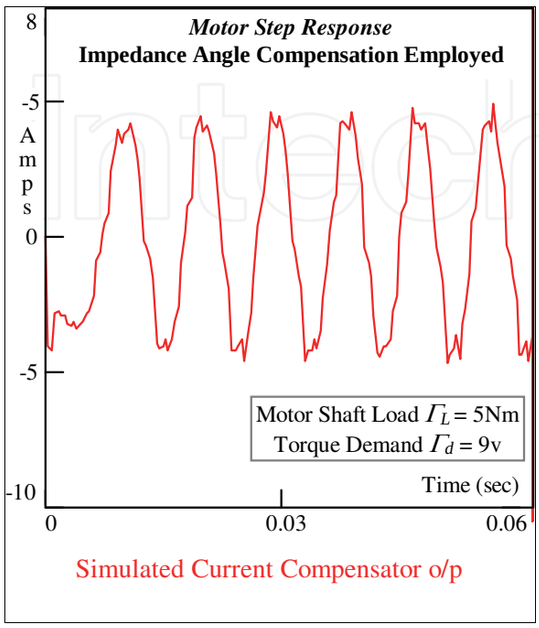


Fig. 39. Motor Current Feedback

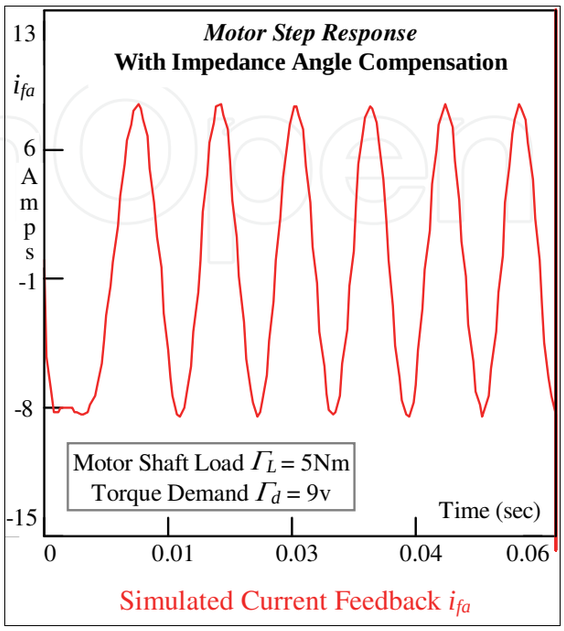


Fig. 40. Current Controller o/p

The simulated stator winding current along with current feedback response and current controller o/p are displayed in Figures 35, 36 and 37 respectively for $\Gamma_d = 5\text{v}$ without saturation related distortion in the current compensator o/p. The BLMD model simulation current characteristics corresponding $\Gamma_d = 9\text{v}$ are also shown in Figures 38, 39 and 40 without saturation effects.

4. BLMD reference model simulation in velocity control mode

In this section the BLMD reference model performance as an ASD emulator is examined and compared with experimental step response data for shaft inertial load conditions with $J_l \sim 3J_m$. Adjustable speed drive operation, with embedded inner PWM current control, is effected by closing the outer velocity loop via a two term PI term controller G_v as shown in Figures 1 and 5. The analog velocity controller shown in Figure 41, which has an inbuilt velocity offset adjustment and speed gain control adjustment K_s for the chosen BLMD system modelled here (Moog GmbH, 1989) has a transfer function

$$G_v(s) = K_e\left(K_p + \frac{K_I}{s}\right) \tag{XXII}$$

with proportional and integral compensation gain settings K_p and K_I respectively. The inclusion of this outer loop velocity compensator, in addition to the inner torque control current loop, results in a complete holistic BLMD reference model that can now be used for ASD simulation and performance evaluation in embedded applications. Proportional and

integral control is easily incorporated in C-language routine during BLMD simulation of velocity closed loop operation as a digital filter code module via (LXV) of the previous chapter using the backward Euler method in (LXIV) of the previous chapter. The resulting ASD model was exercised at low and high shaft velocities corresponding to 36.4% and 73.6% of rated motor speed n_o in Table 1 of the previous chapter and compared with experimental test data at critical internal nodes in Figure 1 for model validation and simulation accuracy.

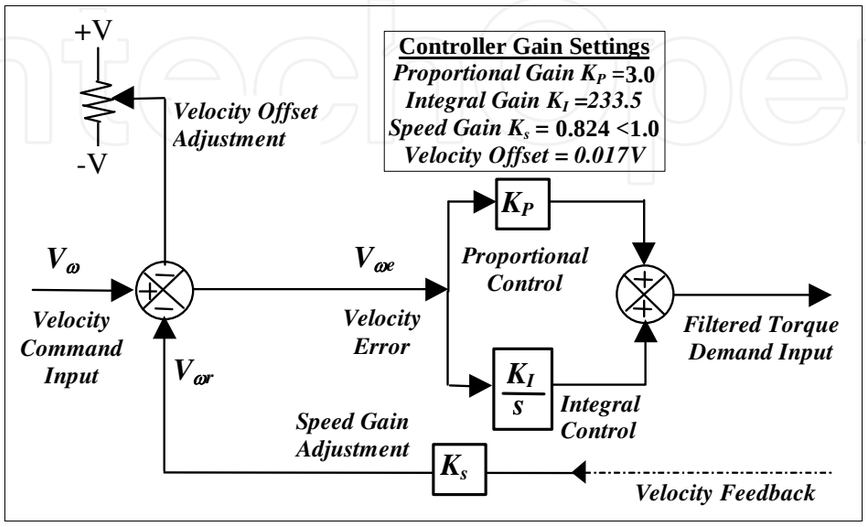


Fig. 41. BLMD Velocity Controller

The current controller G_I step response simulation traces for a velocity step command input V_ω of 2volts, corresponding to 36% of rated motor speed (~4000rpm), are exhibited in Figures 42 to 44 for linear pulsewidth modulator operation.

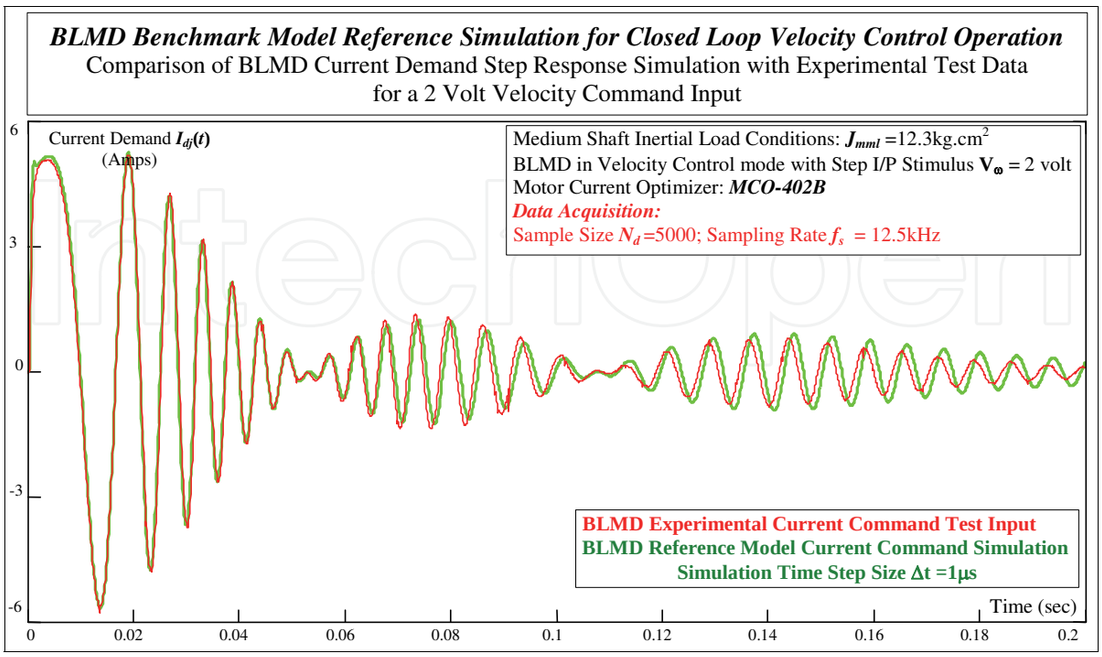


Fig. 42. ASD Reference Model Current Demand Comparison with Experimental Test Data

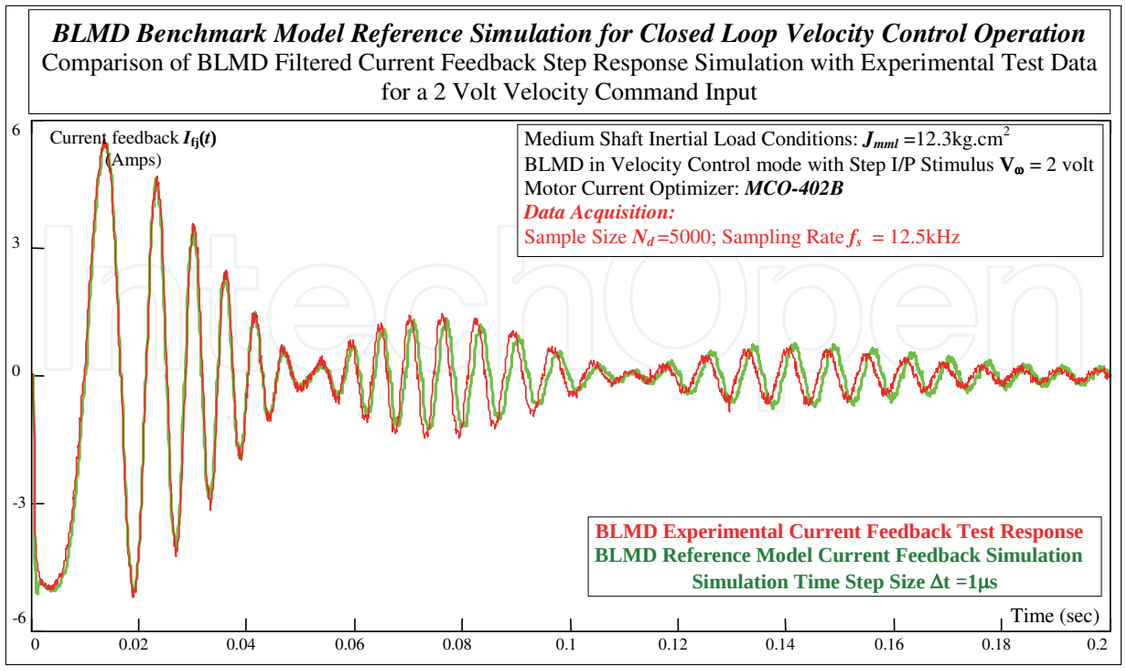


Fig. 43. ASD Model Reference Current Feedback Comparison with Experimental Test Data

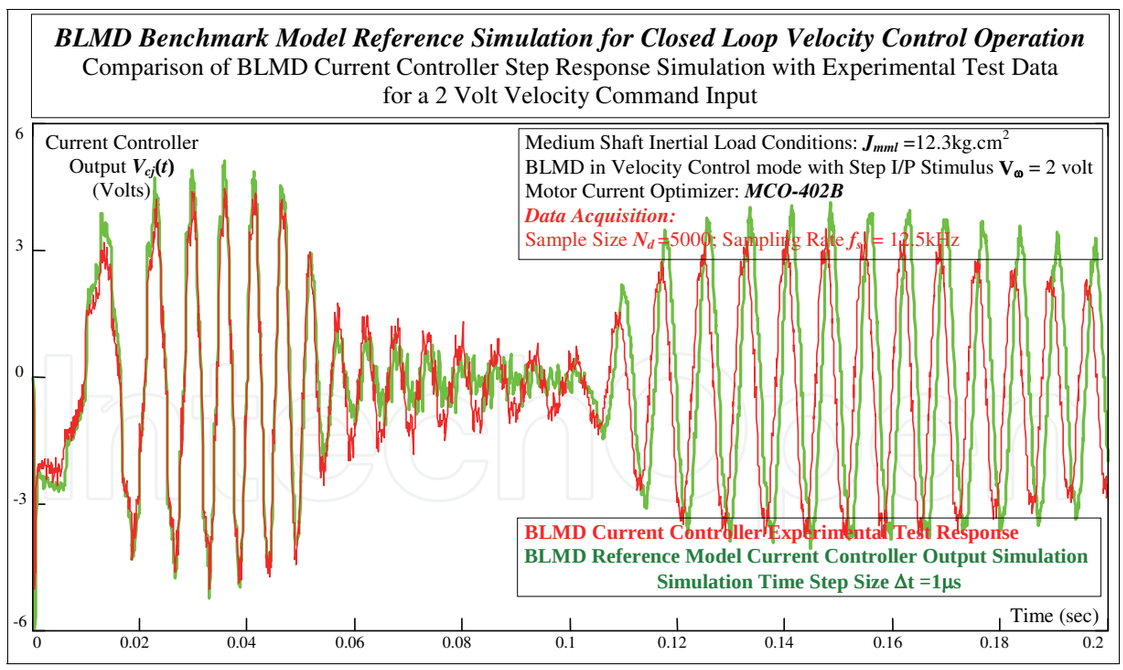


Fig. 44. ASD Model Current Controller o/p Comparison with Experimental Test Data

The accuracy of these simulation traces, which capture the essence of the velocity transient step response $V_{\omega r}$ overshoot in Figure 45, is characterized by a large waveform correlation coefficient of fit in Table V which provides a good indication of the model fidelity when matched with experimental data.

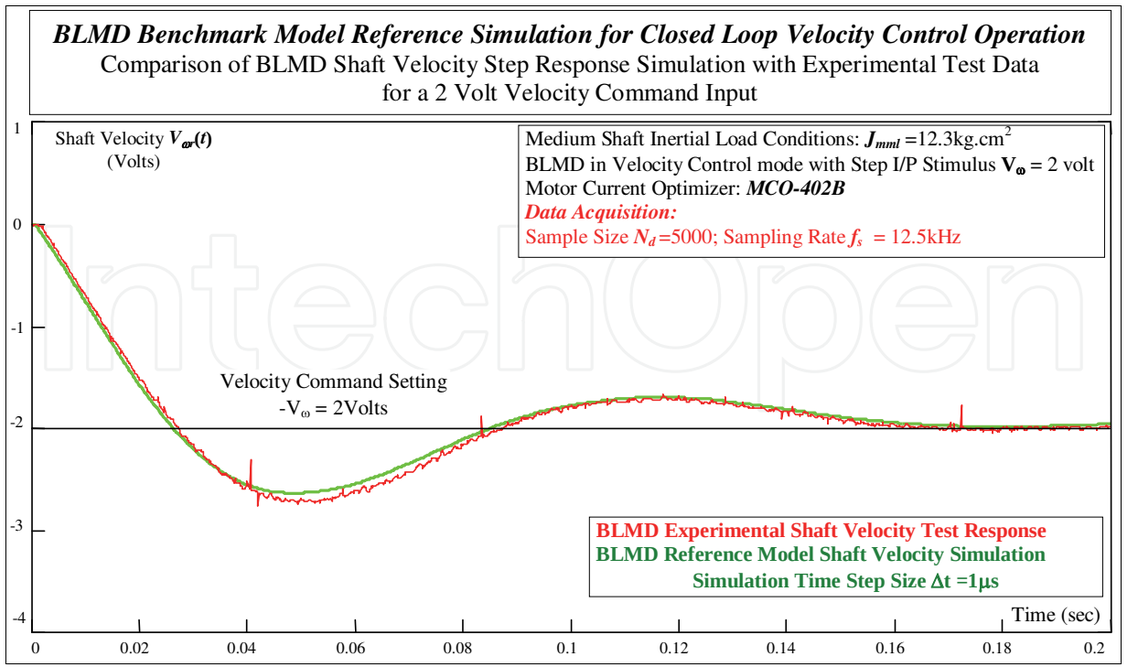


Fig. 45. ASD Model Reference Shaft Velocity Comparison with Experimental Test Data

Target Data Length N_D =3000	Data Sampling Rate: 12.5kHz	BLMD Simulation Time Step: 1μs
Waveform Correlation Analysis for Total Inertial Shaft Load $J_{Tot} = J_l + J_m$. =12.3kg.cm^2		
ASD Waveform	Velocity Command i/p V_ω = 2V	Velocity Command i/p V_ω = 4V
Current Demand I_{dj}	96.8%	92.85%
Current Feedback (FC) I_{fj}	97.26%	93.27%
Current Compensator output V_{cj}	59.81%	45.46%
Motor Shaft Velocity output $V_{\omega r}$	99.8%	99.68%

Table V. ASD Model Trace Simulation Comparison with Experimental Test Data

The simulated current demand and feedback waveforms, which have high matching coefficients with test data, exhibit an amplitude modulated step response with velocity transient overshoot and ringing, before eventually setting to negligible constant amplitude traces with fixed frequency commensurate with reached shaft speed ω_r demanded ($V_{\omega r}$ ~2V) in Figure 45.

The compensated velocity error output for 2Volts operation $V_{\omega r}$ in the BLMD network structure in Figure 5 is equivalent to the filtered torque demand Γ_{df}, as the velocity control effort $\mathbf{V}_{\omega e}$ shown in Figure 46, applied to the inner closed loop for motor current control and BLMD output torque regulation. This optimized velocity error $\mathbf{V}_{\omega e}$ in Figure 46 is a short duration pulse for reasons of fast BLMD shaft velocity risetime T_{res} and short setting time T_{setl} as required in high performance ASD industrial applications.

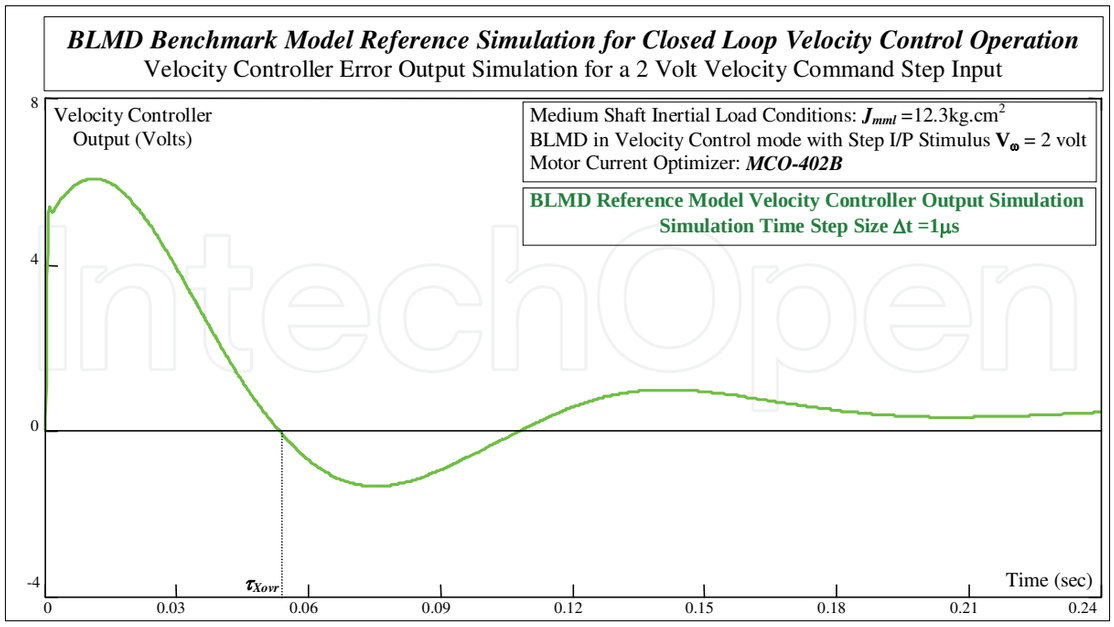


Fig. 46. ASD Reference Model Compensated Velocity Error Output

The presence of overshoot in the BLMD velocity step response in Figure 45 is due to the non-optimal tuning of the velocity controller PI parameters required to ensure stiff dynamical operation for the total drive shaft inertial load $J_{Tot} = J_m + J_l$ in question.

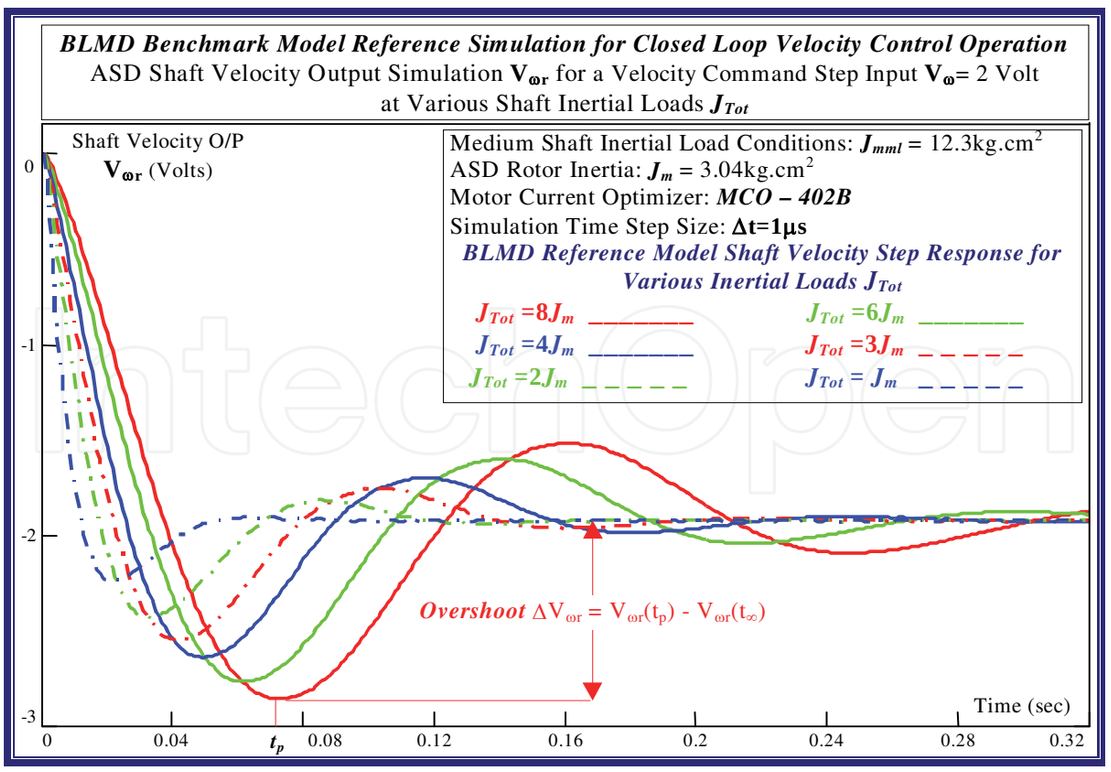


Fig. 47. ASD Shaft Velocity Step Response Variation with Rotor Inertial Load J_{Tot}

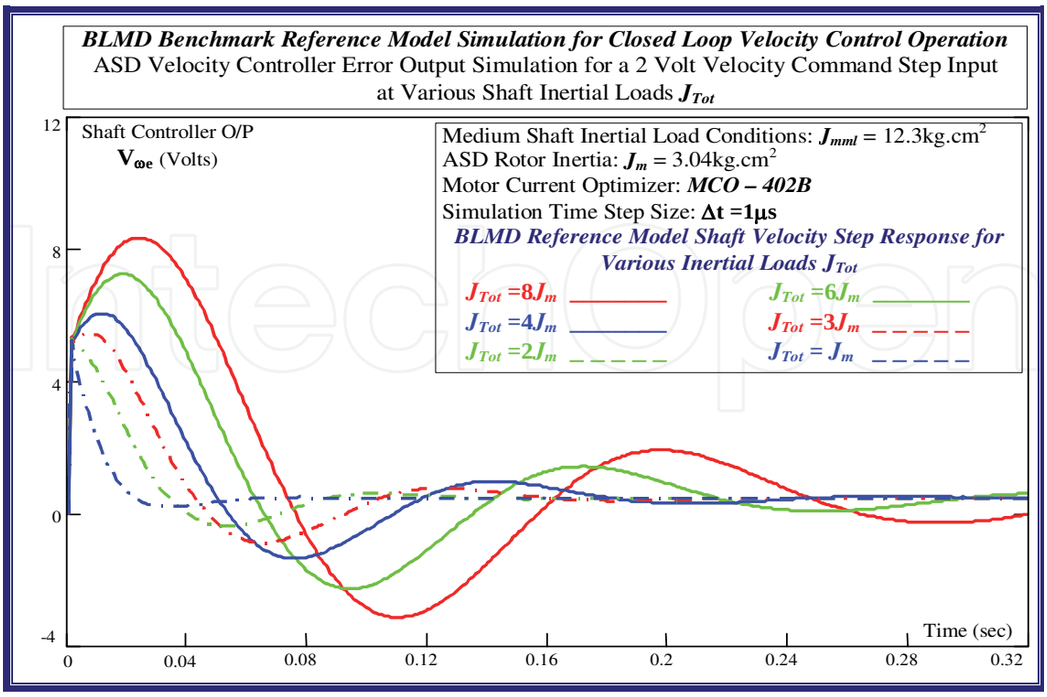


Fig. 48. Variation of ASD Velocity Control Effort with Rotor Inertial Load J_{Tot}

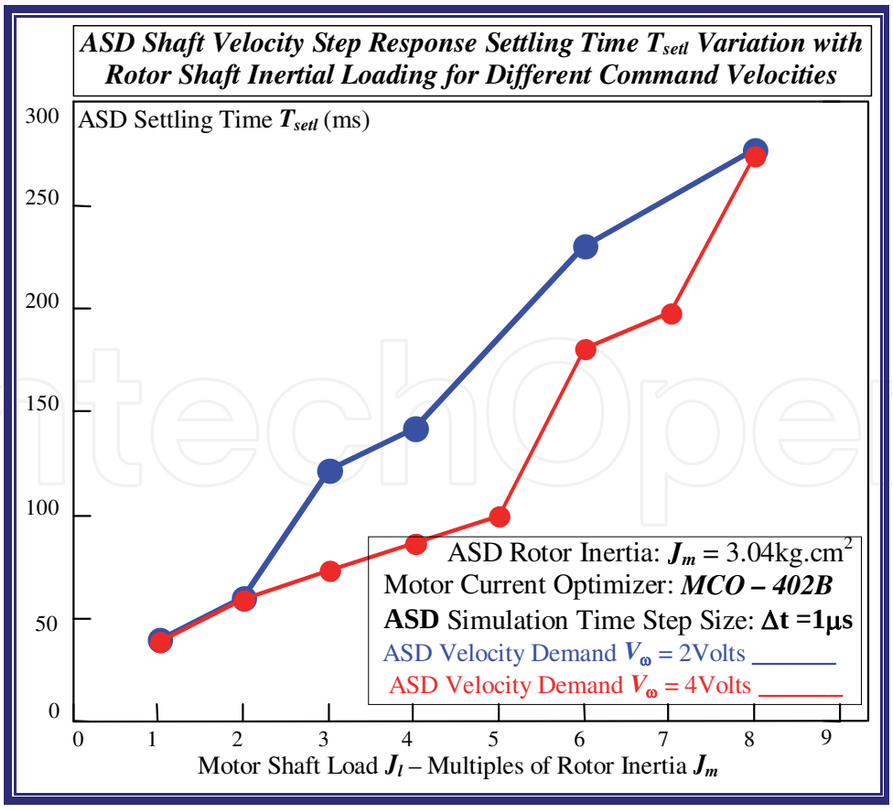


Fig. 49. Variation of ASD Settling Time with Inertial Loading

Examination of the ASD velocity step response trace simulations over a range of shaft inertial load multiples of the rotor value J_m in Figure 47 reveal that the PI parameters have been optimized only at zero load with $J_{Tot} = J_m$ for good drive dynamic transient performance with little overshoot. The effect of increased shaft inertia on the velocity control effort $V_{\omega e}$ in Figure 48, for a velocity command input of 2 volts, is a greater sustained oscillation accompanied by longer settling times T_{setl} manifested in the simulated ASD velocity step response as shown in Figure 49. This behaviour is mirrored by an increased overshoot, as defined in Figure 47, in the BLMD shaft velocity step response with shaft load inertia as shown in Figure 50.
The effect of increased load inertia on ASD dynamic performance also translates into slower rise times T_{res} as shown in Figure 51 for a non optimally tuned velocity controller.
Further ASD step response simulation and comparison with experimental measurements in Figures 52 to 55, for a 4 volts velocity command input which corresponds to 74% of rated motor speed n_0 with resulting saturated pulsewidth modulator operation in Figure 54 for the load inertia considered (~12.3kg.cm²), reveal good BLMD model accuracy.

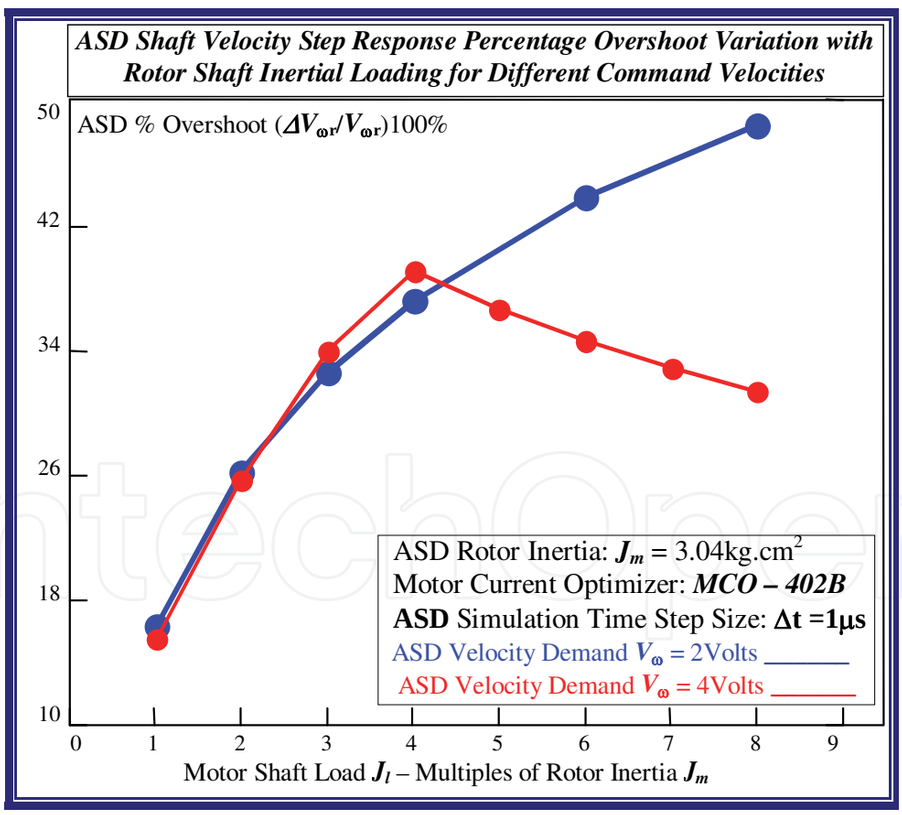


Fig. 50. Variation of ASD Shaft Velocity Overshoot with Inertial Loading

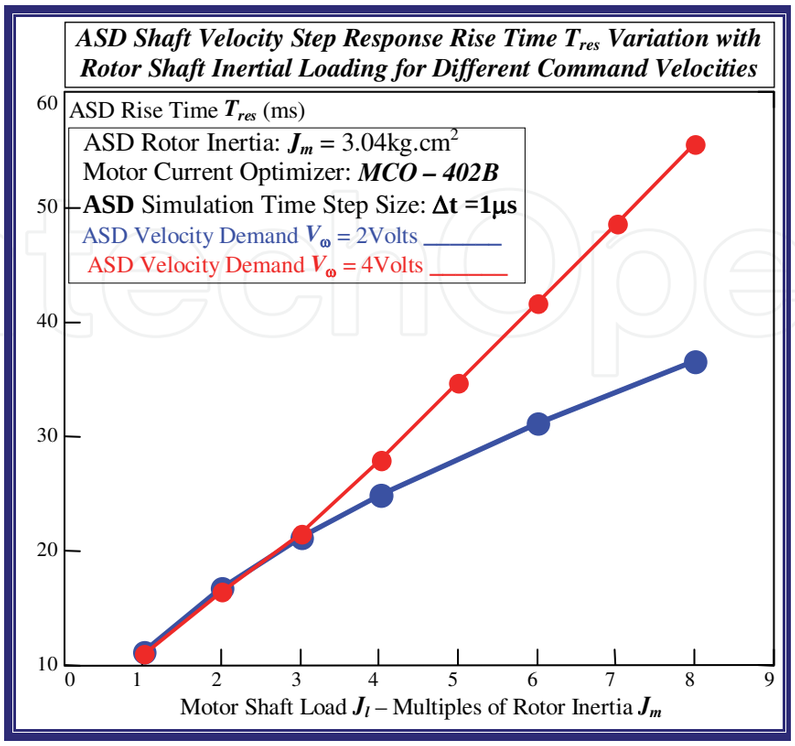


Fig. 51. Variation of ASD Velocity Response Rise Time with Load Inertia

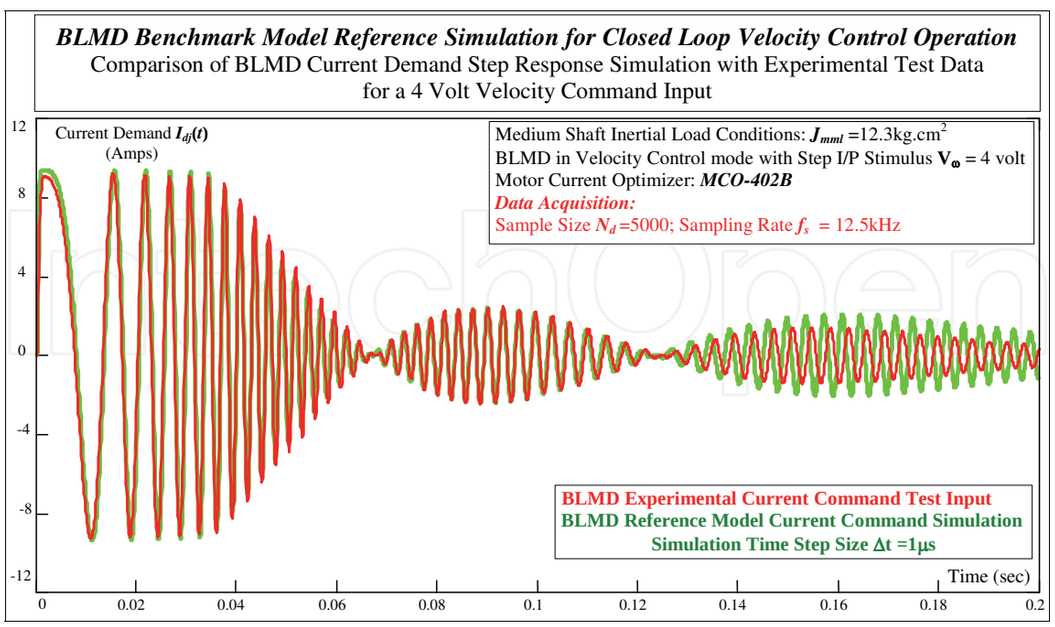


Fig. 52. BLMD Reference Model Current Demand Comparison with Experimental Test Data

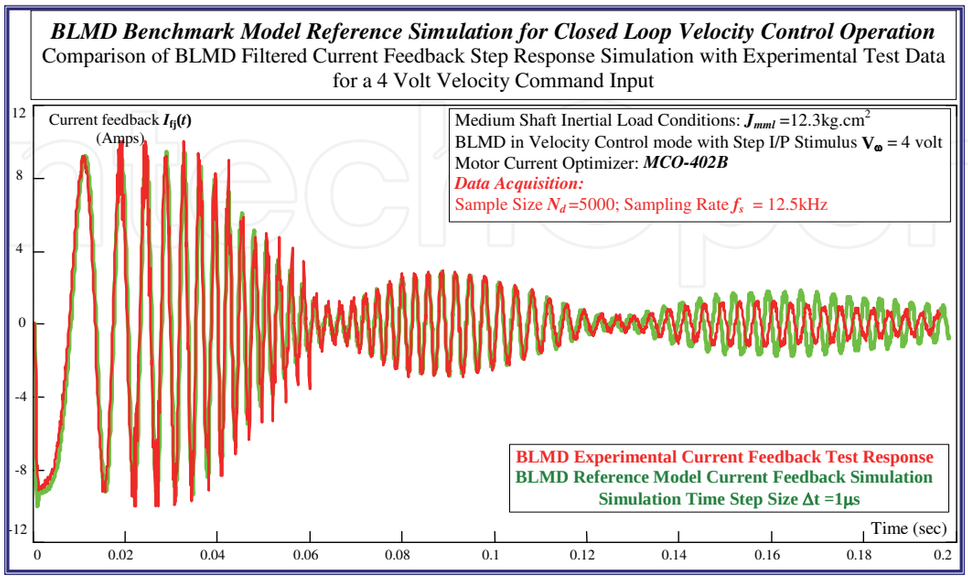


Fig. 53. BLMD Model Current Feedback Comparison with Experimental Test Data

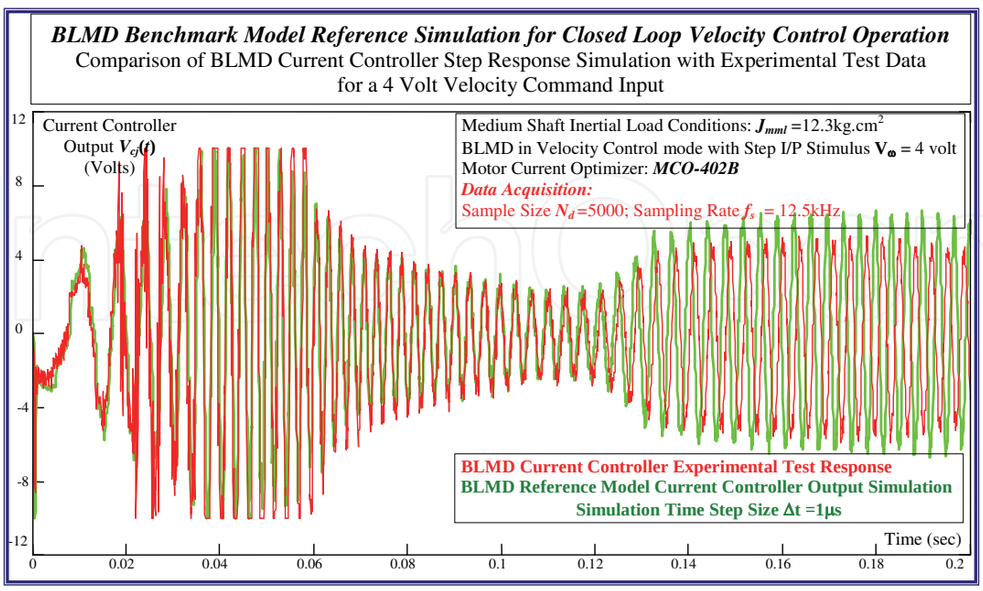


Fig. 54. BLMD Model Current Controller Output Comparison with Experimental Test Data

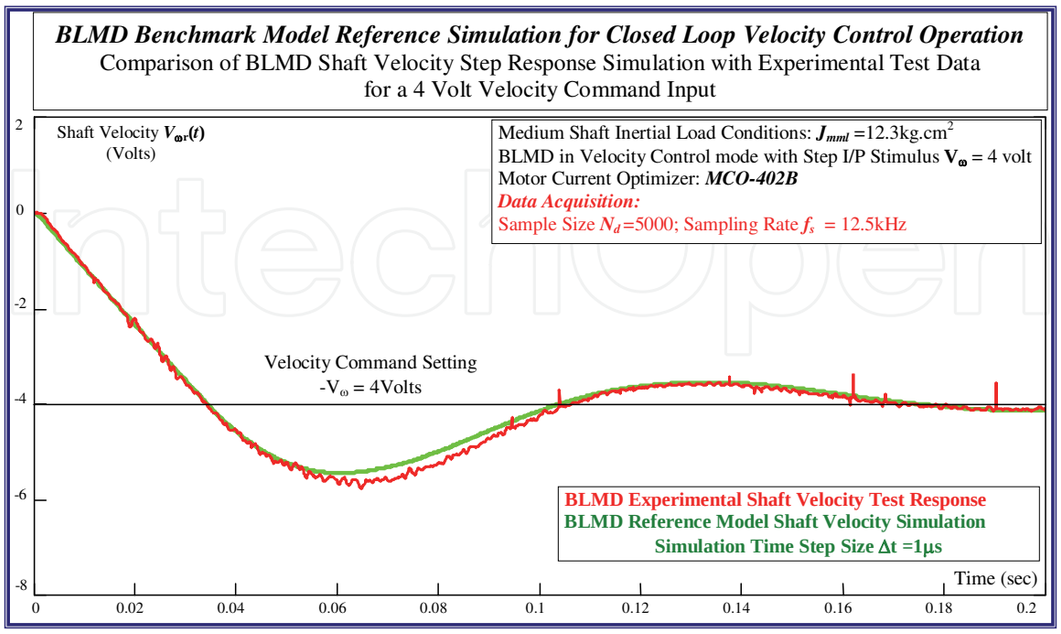


Fig. 55. BLMD Model Shaft Velocity Comparison with Experimental Test Data

The quality of ASD simulation trace match with test data is indicated by the high value of the waveform correlation coefficients given in Table V.

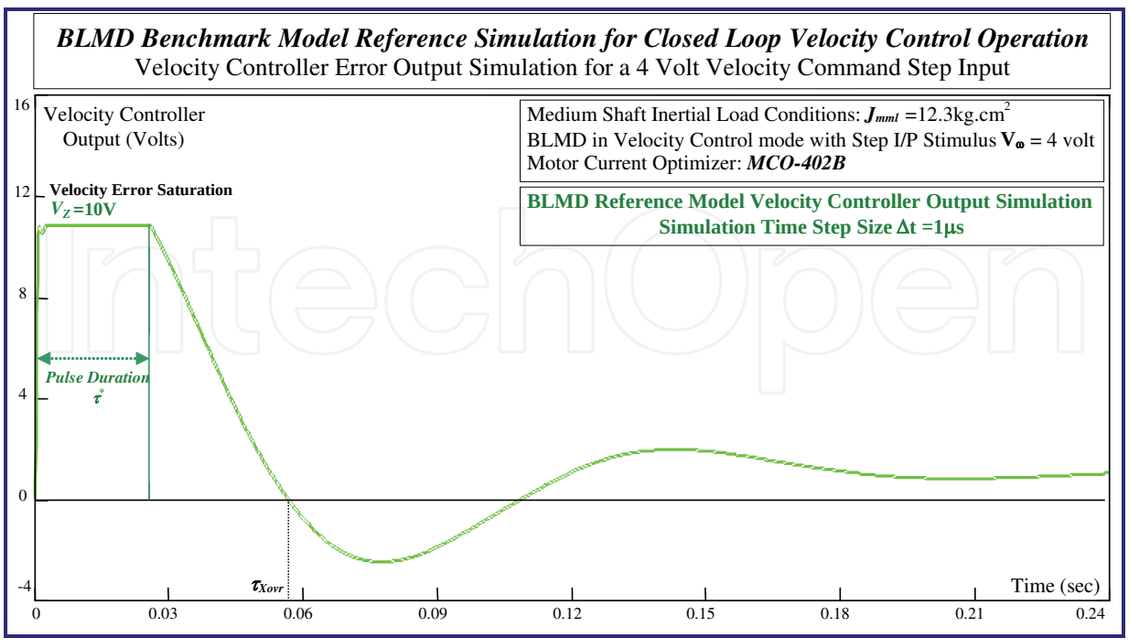


Fig. 56. BLMD Reference Model Compensated Velocity Error Output

The velocity loop derived torque command input stimulus $V_{\omega e}$ in Figure 56 has a pulse duration that is much shorter than the time constant ($\tau_{ml} \sim J_{tot}/B_m$) of the load dynamics, eventhough the pulse amplitude is of sufficient strength to force the shaft velocity to the value demanded (V_{ω}~4volts). The endurance τ^* of the velocity control effort in Figure 56, associated with pronounced PWM saturation in Figure 54, is a measure of the maximum sustained EM torque necessary to accelerate the total BLMD inertial load masses to the appropriate shaft velocity demanded by the ASD command setting V_{ω}. This velocity error pulse has amplitude that is clipped to a maximum saturation limit of 10 volts at the three phase current generator input, which limits the size of the torque loop input stimulus, in the derivation of the BLMD current command signals.

Examination of the family of characteristics pertaining to velocity control effort over a range of motor shaft inertial loads in Figure 57 indicate peak saturation over long pulse intervals τ^* proportional to the inertial masses as in Figure 58 to be accelerated to the required speed $V_{\omega r}$. This velocity error saturation is absent in the characteristics displayed in Figure 48 for 2volt ASD operation and results in linear PWM operation with a BLMD acceleration torque delivery commensurate with the velocity effort.

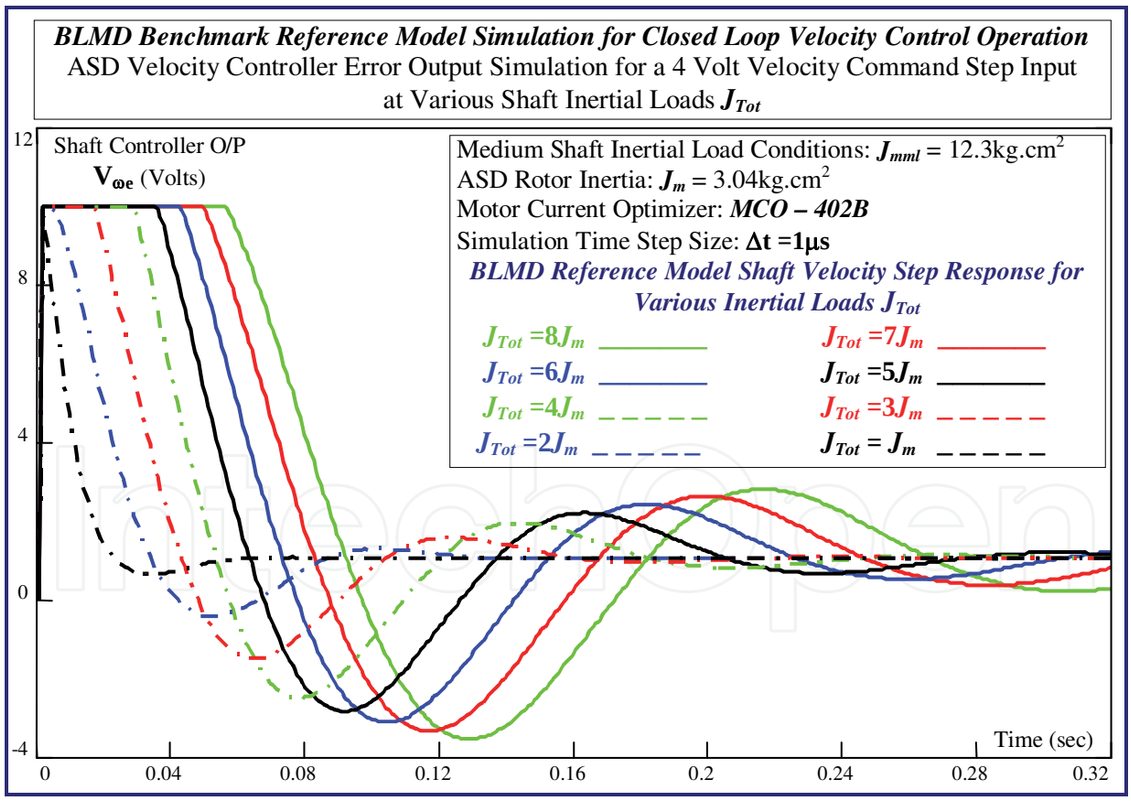


Fig. 57. Variation of Velocity Control Effort with Motor Shaft Inertial Load J_{Tot}

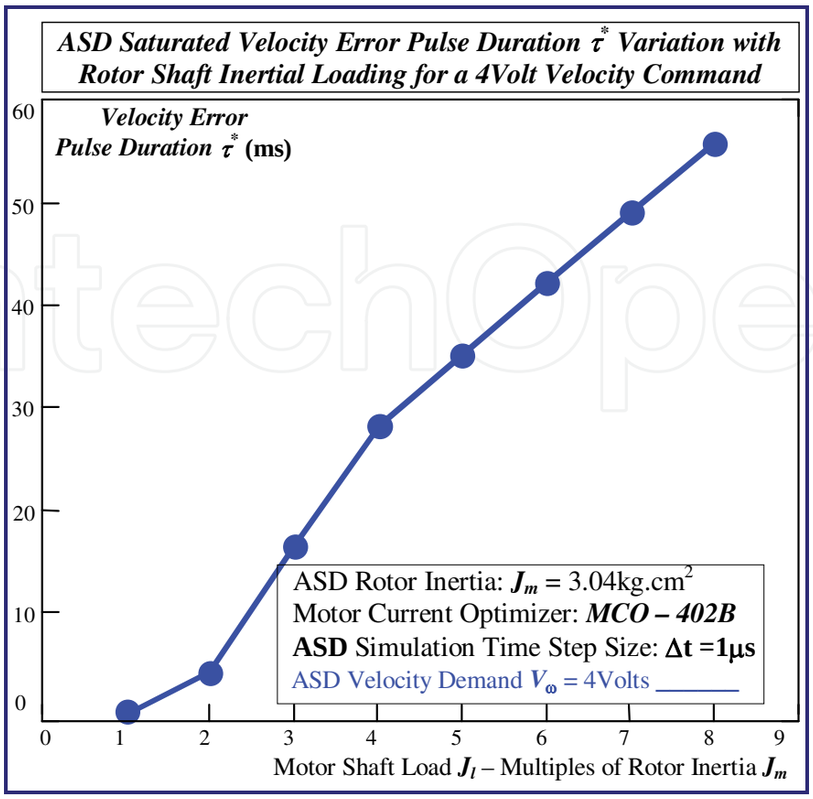


Fig. 58. Variation of Saturated Velocity Error Pulse Duration with Load Inertia

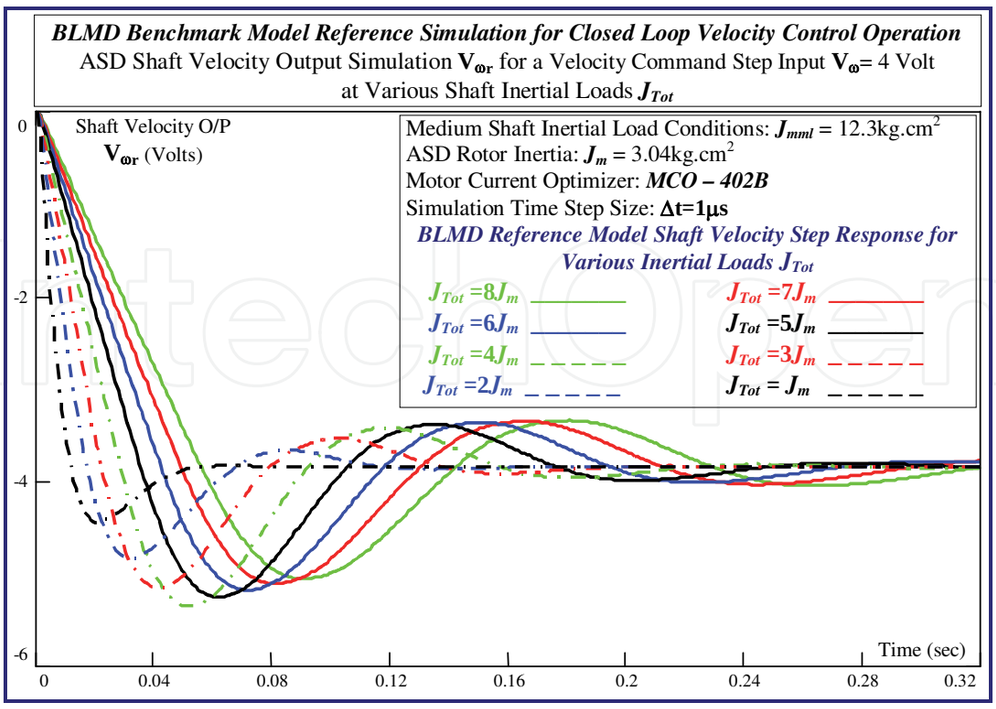


Fig. 59. ASD Velocity Step Response Variation with Inertial Shaft Load

The variation of the simulated ASD velocity step response overshoot and settling time with shaft inertial load for a 4volt velocity command corresponding to 75% of rated shaft speed in Figures 50 and 49 respectively appear to be lower than those for 2 volt ASD operation at 36% rated speed. This is due to the saturation effect of the velocity error in ASD torque generation, which limits the peak amplitude swing of the oscillatory step response on velocity overshoot in Figure 59. However the rise time is longer in this instance as indicated through ASD simulation in Figure 51 for higher velocity command input (~4V) and increases almost linearly with shaft inertial load. The bigger the load inertia being handled during normal ASD operation the greater the corrective action required in the control effort to limit shaft velocity overshoot with inertial deceleration and settling time with reached velocity experienced with a non optimal speed controller during large step changes in velocity command input.
Unless the inertial load is known apriori in this scenario the variable PI parameters cannot be optimally selected for fast risetime and minimum overshoot except through offline manual tuning approximation procedures during the installation and commissioning phase (Moog GmbH, 1988) of the embedded drive in industrial applications. Inertial load parameter extraction in new ASD industrial applications, using the BLMD reference model with step response testing in an offline identification strategy for autotuning purposes, is difficult without knowledge of the initial PI settings of the velocity controller for zero load conditions before drive hookup to the embedded application. In this instance the procedure of accurate inertia parameter extraction using the ASD model in any identification strategy is complicated by the control action of outer velocity loop closure on the wideband inner torque loop when the variable PI settings are unknown. The ASD parameter identification problem is this case has to incorporate evaluation of the existing PI term settings in addition to the inertial load parameter in the new application in order to optimally design the embedded drive velocity controller. However if the drive is configured in torque control mode, thereby eliminating the velocity controller during step response testing in the commissioning phase, inertial load parameters are more amenable to extraction using low torque command input stimuli (<1V) for linear BLMD behaviour with small perturbation of the drive dynamics. The identification problem in this case has to focus only on the extraction of inertial load parameter J_{Tot} and also on the friction coefficient B_m if required.

5. Conclusions

Further BLMD simulation for various load torque settings, based on model confidence, yield results that compare favourably with those obtained from theoretical considerations using catalogued data for the actual drive concerned. The introduction of stator impedance angle compensation at high shaft speed results in improved motor power factor and better BLMD steady state performance. This is verified theoretically and illustrated through model simulation. Detailed BLMD simulation, configuration as an ASD with velocity feedback, is provided at internal observation nodes and checked against measured data at low and high command speed settings for confirmation of model accuracy and validation purposes.

6. Acknowledgment

The author wishes to acknowledge
i. Eolas – The Irish Science and Technology Agency – for research funding.
ii. Moog Ireland Ltd for brushless motor drive equipment for research purposes.

7. References

Balabanian, N. & Bickart, T.A.; *Electrical Network Theory*, 1969, J.Wiley & Sons

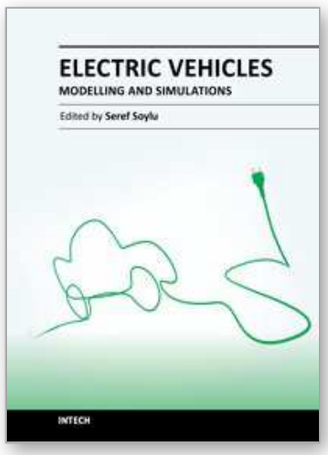
Franklin, G.F. & Powell, J.D.; *Digital Control Of Dynamic Systems*, Addison Wesley, 1980

Guinee, R.A.; (2003). *Modelling, Simulation, and Parameter Identification of a Permanent Magnet Brushless Motor Drive System*, Ph. D. Thesis, NUI – University College Cork.

Guinee, R.A. & Lyden, C.; (1999). Accurate Modelling And Simulation Of A DC Brushless Motor Drive System For High Performance Industrial Applications, IEEE ISCAS'99 - *IEEE International Symposium on Circuits and Systems*, May/June 1999, Orlando, Florida

Moog GmbH; (1988). *Moog Brushless Technology:Brushless Servodrives User Manual*, D310.01.03 En/De/It 01.88, Moog GmbH, D-7030 Böblingen, Germany.

Moog GmbH; (1989). *Moog Brushless Technology User Manual:D31X-XXX Motors,T158-01X Controllers,T157-001 Power Supply*, , D-7030 Böblingen.



Electric Vehicles - Modelling and Simulations

Edited by Dr. Seref Soylu

ISBN 978-953-307-477-1
Hard cover, 466 pages
Publisher InTech
Published online 12, September, 2011
Published in print edition September, 2011

In this book, modeling and simulation of electric vehicles and their components have been emphasized chapter by chapter with valuable contribution of many researchers who work on both technical and regulatory sides of the field. Mathematical models for electrical vehicles and their components were introduced and merged together to make this book a guide for industry, academia and policy makers.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen