```
// ***** TI IQMath CUR_MOD_MACRO and CUR_CONST_MACRO Macros (reference only)
************************************************************

    //From: /home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
controlSUITE_ref/v4.3/cur_mod.h


//File name:       CUR_MOD.H
//
=============================================================================
*/
//#ifndef __CUR_MOD_H__
//#define __CUR_MOD_H__
//
//typedef struct  { _iq  IDs;         // Input: Syn. rotating d-axis current (pu)
//                  _iq  IQs;         // Input: Syn. rotating q-axis current (pu)
//                  _iq  Wr;          // Input: Rotor electrically angular velocity
(pu)
//                  _iq  IMDs;        // Variable: Syn. rotating d-axis magnetizing
current (pu)
//                  _iq  Theta;       // Output: Rotor flux angle (pu)
//                  _iq  Kr;          // Parameter: constant using in magnetizing
current calculation
//                  _iq  Kt;          // Parameter: constant using in slip
calculation
//                  _iq  K;           // Parameter: constant using in rotor flux
angle calculation
//                  _iq  Wslip;       // Variable: Slip
//                  _iq  We;          // Variable: Angular freq of the stator
//                } CURMOD;
//
//
///*-------------------------------------------------------------------------
//Default initalizer for the CURMOD object.
//-------------------------------------------------------------------------*/
//#define CURMOD_DEFAULTS { 0,0,0,0,0, \
//                          0,0,0,0,0 \
//                        }
//
///*-------------------------------------------------------------------------
// CUR_MOD Macro Definition
//-------------------------------------------------------------------------*/
//
//
//#define CUR_MOD_MACRO(v)                                \
//    v.IMDs +=  _IQmpy(v.Kr,(v.IDs - v.IMDs));           \
//    v.Wslip = _IQdiv(_IQmpy(v.Kt,v.IQs),v.IMDs);        \
//    v.We = v.Wr + v.Wslip;                              \
//    v.Theta +=  _IQmpy(v.K,v.We);                       \
//                                                        \
//    if (v.Theta > _IQ(1))                               \
//       v.Theta -=  _IQ(1);                              \
//    else if (v.Theta < _IQ(0))                          \
//       v.Theta += _IQ(1);
//
////v.Theta=(v.Theta+_IQ(1.0))&& 0x00ffffff;
//
```

```
//#endif


    //From: /home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
controlSUITE_ref/v4.3/cur_const.h

///*
=============================================================================
//File name:       CUR_CONST.H
//
=============================================================================
*/
//#ifndef __CUR_CONST_H__
//#define __CUR_CONST_H__
//
//typedef struct  { float32  Rr;          // Input: Rotor resistance (ohm)
//                  float32  Lr;          // Input: Rotor inductance (H)
//                  float32  fb;          // Input: Base electrical frequency (Hz)
//                  float32  Ts;          // Input: Sampling period (sec)
//                  float32  Kr;          // Output: constant using in magnetizing
current calculation
//                  float32  Kt;          // Output: constant using in slip
calculation
//                  float32  K;           // Output: constant using in rotor flux
angle calculation
//                  float32  Tr;          // Variable: Rotor time constant (sec)
//                } CURMOD_CONST;
//
//
///*-----------------------------------------------------------------------------
//Default initalizer for the CURMOD_CONST object.
//-----------------------------------------------------------------------------*/
//#define CURMOD_CONST_DEFAULTS { 0,0,0,0, \
//                                0,0,0,0  \
//                              }
//
///*-----------------------------------------------------------------------------
// CUR_CONST Macro Definition
//-----------------------------------------------------------------------------*/
//
//#define PI 3.14159265358979
//
//
//#define CUR_CONST_MACRO(v)      \
//    v.Tr = v.Lr/v.Rr;           \
//                                \
//    v.Kr = v.Ts/v.Tr;           \
//    v.Kt = 1/(v.Tr*2*PI*v.fb);  \
//    v.K = v.Ts*v.fb;
//
//#endif

    //From: /home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
controlSUITE_ref/HVACI_Sensored/HVACI_Sensored.c

//// Initialize the CUR_MOD constant module
//    cm1_const.Rr = RR;
//    cm1_const.Lr = LR;
//    cm1_const.fb = BASE_FREQ;
```

```
//    cm1_const.Ts = T;
//    CUR_CONST_MACRO(cm1_const)

//
*******************************************************************************
*****************************************************


// ***** TI IQMath PI_MACRO  Macro (reference only)
*******************************************************************************
*********

    //From: /home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
controlSUITE_ref/v4.3/pi.h


///*
================================================================================
//File name:       PI.H
//
================================================================================
*/
//
//
//#ifndef __PI_H__
//#define __PI_H__
//
//typedef struct {  _iq  Ref;             // Input: reference set-point
//                  _iq  Fbk;             // Input: feedback
//                  _iq  Out;             // Output: controller output
//                  _iq  Kp;              // Parameter: proportional loop gain
//                  _iq  Ki;              // Parameter: integral gain
//                  _iq  Umax;            // Parameter: upper saturation limit
//                  _iq  Umin;            // Parameter: lower saturation limit
//                  _iq  up;              // Data: proportional term
//                  _iq  ui;              // Data: integral term
//                  _iq  v1;              // Data: pre-saturated controller output
//                  _iq  i1;              // Data: integrator storage: ui(k-1)
//                  _iq  w1;              // Data: saturation record: [u(k-1) -
v(k-1)]
//               } PI_CONTROLLER;
//
//
///*-----------------------------------------------------------------------------
//Default initalisation values for the PI_GRANDO objects
//-----------------------------------------------------------------------------*/
//
//#define PI_CONTROLLER_DEFAULTS {        \
//                           0,           \
//                           0,           \
//                           0,           \
//                           _IQ(1.0),    \
//                           _IQ(0.0),    \
//                           _IQ(1.0),    \
//                           _IQ(-1.0),   \
//                           _IQ(0.0),    \
//                           _IQ(0.0),    \
//                           _IQ(0.0),    \
//                           _IQ(0.0),    \
```

```
//                          _IQ(1.0)     \
//                        }
//
///*-----------------------------------------------------------------------------
//    PI_GRANDO Macro Definition
//-----------------------------------------------------------------------------*/
//
//#define PI_MACRO(v)                                                   \
//                                                                      \
//    /* proportional term */                                           \
//    v.up = _IQmpy(v.Kp, (v.Ref - v.Fbk));                             \
//                                                                      \
//    /* integral term */                                               \
//    v.ui = (v.Out == v.v1)?(_IQmpy(v.Ki, v.up)+ v.i1) : v.i1;         \
//    v.i1 = v.ui;                                                      \
//                                                                      \
//    /* control output */                                              \
//    v.v1 = v.up + v.ui;                                               \
//    v.Out= _IQsat(v.v1, v.Umax, v.Umin);                              \
//    //v.w1 = (v.Out == v.v1) ? _IQ(1.0) : _IQ(0.0);                   \
//
////
*******************************************************************************
////   This macro works with angles as inputs, hence error is rolled within -pi to
+pi
////
*******************************************************************************
//#define PI_POS_MACRO(v)                                               \
//    /* proportional term */                                           \
//    v.up = v.Ref - v.Fbk;                                             \
//    if (v.up >= _IQ(0.5))                                             \
//      v.up -= _IQ(1.0);             /* roll in the error */           \
//    else if (v.up <= _IQ(-0.5))                                       \
//      v.up += _IQ(1.0);             /* roll in the error */           \
//                                                                      \
//    /* integral term */                                               \
//    v.up = _IQmpy(v.Kp, v.up);                                        \
//    v.ui = (v.Out == v.v1)?(_IQmpy(v.Ki, v.up)+ v.i1) : v.i1;         \
//    v.i1 = v.ui;                                                      \
//                                                                      \
//    /* control output */                                              \
//    v.v1 = v.up + v.ui;                                               \
//    v.Out= _IQsat(v.v1, v.Umax, v.Umin);                              \
//    //v.w1 = (v.Out == v.v1) ? _IQ(1.0) : _IQ(0.0);                   \
//
//
//#endif // __PI_H__

//
*******************************************************************************
*************************************************


// ***** TI IQMath CLARKE_MACRO Macro (reference only)
*******************************************************************************
******


//From: /home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
```

controlSUITE_ref/v4.3/clarke.h

```
///*
=============================================================================
//File name:       CLARKE.H
//
=============================================================================
*/
//
//
//#ifndef __CLARKE_H__
//#define __CLARKE_H__
//
//typedef struct {  _iq  As;          // Input: phase-a stator variable
//                  _iq  Bs;          // Input: phase-b stator variable
//                  _iq  Cs;          // Input: phase-c stator variable
//                  _iq  Alpha;       // Output: stationary d-axis stator variable
//                  _iq  Beta;        // Output: stationary q-axis stator variable
//               } CLARKE;
//
///*-----------------------------------------------------------------------------
//    Default initalizer for the CLARKE object.
//-----------------------------------------------------------------------------*/
//#define CLARKE_DEFAULTS { 0, \
//                          0, \
//                          0, \
//                          0, \
//                          0, \
//                        }
//
///*-----------------------------------------------------------------------------
//    CLARKE Transformation Macro Definition
//-----------------------------------------------------------------------------*/
//
////  1/sqrt(3) = 0.57735026918963
//#define  ONEbySQRT3   0.57735026918963    /* 1/sqrt(3) */
//
//
//// Clarke transform macro (with 2 currents)
////=========================================
//#define CLARKE_MACRO(v)                                         \
//v.Alpha = v.As;                                                 \
//v.Beta = _IQmpy((v.As +_IQmpy2(v.Bs)),_IQ(ONEbySQRT3));
//
//
//// Clarke transform macro (with 3 currents)
////=========================================
//#define CLARKE1_MACRO(v)                                        \
//v.Alpha = v.As;                                                 \
//v.Beta  = _IQmpy((v.Bs - v.Cs),_IQ(ONEbySQRT3));
//
//#endif // __CLARKE_H__

//
*******************************************************************************
*****************************************************


// ***** TI IQMath PARK_MACRO Macro (reference only)
```

```
*********************************************************************************
********


        //From:
/home/maiello/Desktop/Development-Simulation/InductionMotor_DTC_SVM/
controlSUITE_ref/v4.3/park.h

///*
=================================================================================
//File name:       PARK.H
//
=================================================================================
*/
//
//#ifndef __PARK_H__
//#define __PARK_H__
//
//typedef struct {  _iq  Alpha;       // Input: stationary d-axis stator variable
//                  _iq  Beta;        // Input: stationary q-axis stator variable
//                  _iq  Angle;       // Input: rotating angle (pu)
//                  _iq  Ds;          // Output: rotating d-axis stator variable
//                  _iq  Qs;          // Output: rotating q-axis stator variable
//                  _iq  Sine;
//                  _iq  Cosine;
//                } PARK;
//
///*-------------------------------------------------------------------------------
//Default initalizer for the PARK object.
//-------------------------------------------------------------------------------*/
//#define PARK_DEFAULTS {   0, \
//                          0, \
//                          0, \
//                          0, \
//                          0, \
//                          0, \
//                          0, \
//                          }
//
///*-------------------------------------------------------------------------------
//    PARK Transformation Macro Definition
//-------------------------------------------------------------------------------*/
//
//
//#define PARK_MACRO(v)                                                   \
//                                                                        \
//    v.Ds = _IQmpy(v.Alpha,v.Cosine) + _IQmpy(v.Beta,v.Sine);    \
//    v.Qs = _IQmpy(v.Beta,v.Cosine) - _IQmpy(v.Alpha,v.Sine);
//
//#endif // __PARK_H__

//
*********************************************************************************
************************************************
```