```
IF ( Initialize ) THEN  ! .TRUE. if we're on the first call to the subroutine
   WRITE(*,*)  'Running with pitch control programmed by Eric Anderson '// &
              'in subroutine PitchCntrl(), which can be found in UserSubs.f90 '
    Initialize = .FALSE.
   ! Initialize the SAVEd variables:
   PitCom     = BlPitch                                ! This will ensure that the variable
speed controller picks the correct control region and the pitch controller pickes the
correct gain on the first call. If pitchCtrl()is called before UsrVSCtrl()
initializing it here will work, if not I need to do something else.
   GK         = 1.0/( 1.0 + PitCom(1)/PC_KK )    ! This will ensure that the pitch
angle is unchanged if the initial SpdErr is zero
   IntSpdErr  = PitCom(1)/( GK*PC_KI )                 ! This will ensure that the pitch
angle is unchanged if the initial SpdErr is zero
   LastTimePC = ZTime - PC_DT                          ! This will ensure that the pitch
controller is called on the first pass
ENDIF

!========================================================================
   ! Pitch control:

   ! Compute the elapsed time since the last call to the controller:
ElapTime = ZTime - LastTimePC

   ! Only perform the control calculations if the elapsed time is greater than
   !   or equal to the communication interval of the pitch controller:
   ! NOTE: Time is scaled by OnePlusEps to ensure that the contoller is called
   !       at every time step when PC_DT = DT, even in the presence of
   !       numerical precision errors.

IF ( ( ZTime*OnePlusEps - LastTimePC ) >= PC_DT )  THEN

    CALL updateControlParameters( HSS_Spd, ZTime )

    IF ( EmergencyShutdown ) THEN
        PitComT = 3.1415926535/2
    ELSE
       ! Compute the gain scheduling correction factor based on the previously
       !   commanded pitch angle for blade 1:
          GK = 1.0/( 1.0 + PitCom(1)/PC_KK )

       ! Compute the current speed error and its integral w.r.t. time;
          SpdErr    = GenSpeedF - PC_RefSpd                                   ! Current
speed error
          IntSpdErr = IntSpdErr + SpdErr*ElapTime                             ! Current
integral of speed error w.r.t. time

        ! saturate the integral term using the pitch angle limits:
          IntSpdErr = MIN( MAX(IntSpdErr,PC_MinPit/(GK*PC_KI)), &
                          PC_MaxPit/( GK*PC_KI))     ! Saturate the integral term using
the pitch angle limits, converted to integral speed error limits

        ! Compute the pitch commands associated with the proportional and integral
gains:
          PitComP    = GK*PC_KP*   SpdErr                                     !
Proportional term
          PitComI    = GK*PC_KI*IntSpdErr                                     ! Integral
term (saturated)


        ! Superimpose the individual commands to get the total pitch command;
          PitComT    = PitComP + PitComI                                      ! Overall
```