

```

LOGICAL, SAVE :: Initialize = .TRUE. !Flag used to
initialize some saved variables on the first call to this subroutine

!=====
! Initialize saved variables on first call to subroutine
IF ( Initialize ) THEN
    WRITE(*,*) 'First call to subroutine updateControlParameters(), '// &
'programmed by Eric Anderson. Subroutine can be found in UserSubs.f90 '
    Initialize = .FALSE.
    GenSpeedF = HSS_Spd ! This will ensure that generator speed filter
will use the initial value of the generator speed on the first pass
    LastTime = ZTime ! This will ensure that generator speed filter
will use the initial value of the generator speed on the first pass
ENDIF
!=====
! Filter the HSS (generator) speed measurement:
! NOTE: This is a very simple recursive, single-pole, low-pass filter with
! exponential smoothing.
! Update the coefficient in the recursive formula based on the elapsed time
! since the last call to the controller:
Alpha = EXP( ( LastTime - ZTime )*CornerFreq )
! Apply the filter:
GenSpeedF = ( 1.0 - Alpha )*HSS_Spd + Alpha*GenSpeedF
!=====
!Derate Calculations
IF (ZTime >= TimeDREnd) THEN
    !return turbine to normal operation
    FF_pwrFactor = 1.0 - DerateFactor + DerateFactor*(1.0 - pDR*(ZTime - &
TimeDREnd)*EXP(-pDR*(ZTime - TimeDREnd)) - EXP(-pDR*(ZTime - TimeDREnd)))
ELSEIF (ZTime >= TimeDRStart) THEN
    !Derate turbine
    FF_pwrFactor = 1.0 - DerateFactor*(1.0 - pDR*(ZTime - TimeDRStart)*&
EXP(-pDR*(ZTime - TimeDRStart)) - EXP(-pDR*(ZTime - TimeDRStart)))
ENDIF
!=====
! Set pitch control parameters
PC_RefSpd = PC_RefSpd_baseline*FF_pwrFactor
DO interpCounter = 2, size(DRPitchArray)
    IF ( (FF_pwrFactor .GT. DRArray(interpCounter-1) ) .AND. &
(FF_pwrFactor .LT. DRArray(interpCounter) ) ) THEN
        PC_MinPit = DRPitchArray(interpCounter-1) + ( DRPitchArray&
(interpCounter) - DRPitchArray(interpCounter-1) )*( FF_pwrFactor - &
DRArray(interpCounter-1) )/( DRArray(interpCounter) - DRArray(interpCounter-1) )
        WRITE(*,*) 'PowerFactor = ',FF_pwrFactor,' PC_MinPit =',PC_MinPit
    ENDIF
ENDDO
!=====
! Set torque control parameters
VS_Rgn2_K = VS_Rgn2K_baseline/(FF_pwrFactor**2) ! Region 2 torque constant
VS_RtPwr = VS_RtPwr_baseline*FF_pwrFactor ! Rated power
!=====
! Check to see if emergency shutdown should be initiated
IF ((ZTime > 30.0) .AND. (EmergencyShutdown .EQV. .FALSE.)) THEN ! If simulation
has run long enough to pass the initial transient behavior and an emergency shutdown
hasn't been requested yet.
    IF ( ( ZTime > TEmShutdown ) .OR. ( (100*(HSS_Spd-PC_RefSpd_baseline)&
/PC_RefSpd_baseline) .GE. maxOverspeed) ) THEN ! Should an emergency
shutdown be requested now?
        EmergencyShutdown = .TRUE.
        WRITE(*,*) 'Emergency shutdown requested at T =',ZTime, &
'Overspeed =',(100*(HSS_Spd-PC_RefSpd_baseline)/PC_RefSpd_baseline)
    
```