KU DESIGN GUIDELINES
APPENDIX XVII
LIBRARY OF STANDARD CONTROL SEQUENCES

GEN-001 – STANDARD PID FUNCTION

This routine is a standard control loop function involving an analog control variable and an analog control device.

```
'PID.F - PRODUCES SIGNAL FROM 0 TO 1 REPRESENTING 0-100% CAPACITY
Arg 1 cv '    controlled variable (sensor)
Arg 2 sp '    setpoint
Arg 3 kp '    proportional gain - constant
Arg 4 ki '    integral gain - constant
Arg 5 kd '    derivative gain - constant
Arg 6 bias '  initial output value when no error exists
Arg 7 actn '  directional relationship of cv and output capacity (1=DA 0=RA)
Arg 8 lerr '  error from previous scan - stored in calling program
Arg 9 intg '  integral value - stored in calling program
Arg 10 lstm ' last time the function was called - datetime variable

Numeric err ' difference between sp and cv
Numeric dltm 'difference between current time and lstm
Numeric op '  output capacity (0-1) - returned to calling program

If (passed(10)) then
  dltm = maximum((Date - lstm), Scan)
  lstm = Date
Else
  dltm = maximum(Scan, 0.01)
Endif

err = ((cv - sp) * (actn = 1)) + ((sp - cv) * (actn = 0))
op = maximum(minimum(((kp * err) + (ki * intg) + (kd * ((err - lerr) / dltm)) + bias), 1), 0)
intg = maximum(intg + (err * dltm * (((op < 1) or (err < 0)) and ((op > 0) or (err > 0)))), 0)
lerr = err
Return (op)
```

EXAMPLE:

This routine provides the capability to provide Proportional-Integral-Derivative (PID) control for a process variable. It can be used for temperature, pressure, and flow control applications.

GEN-002 – STANDARD LEAD/LAG DECISION PROGRAM

This routine is standard logic for a pump lead/lag configuration where the lead is alternated automatically on a regular basis.

```
'Tri-M Standard LEADLAG01

Line DECISION
  If CHWP5.LEAD then Goto CHWP5LEAD
  Goto CHWP6LEAD

Line CHWP5LEAD
  CHWP5.LEAD = On
  CHWP6.LEAD = Off
  If (Weekday = Wednesday) and (Hourofday = 7) then Goto SWITCHTO2

Line SWITCHTO2
  CHWP5.LEAD = Off
  CHWP6.LEAD = On
  If (Weekday <> Wednesday) or (Hourofday >= 8) then Goto CHWP6LEAD

Line CHWP6LEAD
  CHWP5.LEAD = Off
  CHWP6.LEAD = On
  If (Weekday = Wednesday) and (Hourofday = 7) then Goto SWITCHTO1

Line SWITCHTO1
  CHWP5.LEAD = On
  CHWP6.LEAD = Off
  If (Weekday <> Wednesday) or (Hourofday >= 8) then Goto CHWP5LEAD

Line E
  LEADLAG.E.TIME = Date
  Goto DECISION
```

Example:
When two identical pumps, such as heating hot water pumps, are installed, this program is used to automatically alternate the lead pump at a specific time on a specific day of the week.

HVAC-001 – STANDARD ANALOG PID CONTROL PROGRAM

This is a standard control loop function for temperature control via a modulating valve.

Required External Software Numeric Points:
OUTPUT.rhi, OUTPUT.rlo, OUTPUT.PID

Program Definition:
Numeric actn, CO, OUTPUTkp, OUTPUTki, OUTPUTkd, OUTPUTdbnd, OUTPUTlerr

```
INITIALIZE:
  actn = -1
  OUTPUTkp = 0.1
  OUTPUTki = 0.01
  OUTPUTkd = 0.001
  OUTPUTdbnd = 0.2 'Deg F
  OUTPUT.rhi = 75
  OUTPUT.rlo = 55
  Goto CNTRL

CNTRL:
  If ((abs(CONTROL.POINT - CONTROL.POINT.SETPOINT)) < OUTPUTdbnd) then
Goto IN.DeadBand
  PID.F(CONTROL.POINT, CONTROL.POINT.SETPOINT, OUTPUTkp, OUTPUTki,
OUTPUTkd, actn, OUTPUTlerr, OUTPUTint, CO)
  OUTPUT.PID = CO
  OUTPUT = ((OUTPUT.PID * (OUTPUT.rhi - OUTPUT.rlo)) + OUTPUT.rlo)
  Goto MODU.WAIT

MODU.WAIT:
  If (TS >= 2) then Goto CNTRL

IN.DeadBand:
  If ((abs(CONTROL.POINT - CONTROL.POINT.SP)) > OUTPUTdbnd) or (TS > 60)
then Goto CNTRL
```

Variable Definitions:
actn = action (1=DA/-1=RA)
kp = proportional gain
ki = integral gain
kd = derivative
dbnd = deadband
lerr = error
int = integral value

rhi = control range high value
rlo = control range low value
CO = PID function returned value
OUTPUT = controlled variable
CONTROL.POINT = reference variable
CONTROL.POINT.SETPOINT = reference variable setpoint value
PID.F = PID function call

Example:
Analog heating valve control – as supply air temperature falls below its active setpoint, the heating valve will modulate open. As the supply air temperature rises above its active setpoint, the heating valve will modulate closed.

HVAC-002 – **STANDARD AIR HANDLER MODE PROGRAM**

This logic sets the HVAC control mode to Heating, Economizer, or Mechanical Cooling.

```
Line Decision
  If MCLG then Goto MECH_MODE
  Goto HEAT_ECON_MODE

Line HEAT_ECON_MODE
  HTG = On
  ECON = On
  MCLG = Off
  AHU.MODE.S = "Heating/Economizer"
  If (not CHWAVAIL) then Goto HEAT_ECON_MODE
  If (OAT > OAT.HTG.SP) and ((ST > (CLG.TRG - ((CLG.TRG - HTG.TRG) / 2))) and
(RHCV <= 0)) then Goto MECH_WAIT

Line MECH_WAIT
  HTG = On
  ECON = On
  MCLG = Off
  AHU.MODE.S = "Heating/Economizer"
  If (not CHWAVAIL) or (OAT < OAT.HTG.SP) or (ST < (CLG.TRG - ((CLG.TRG -
HTG.TRG) / 2))) or (RHCV > 0)
  then Goto HEAT_ECON_MODE
  If (TM >= 10) then Goto MECH_MODE

Line MECH_MODE
  HTG = Off
  ECON = Off
  MCLG = On
  AHU.MODE.S = "Mechanical Cooling"
  If (not CHWAVAIL) then Goto HEAT_ECON_MODE
  If (OAT < (OAT.HTG.SP - 2)) or ((ST < (HTG.TRG + ((CLG.TRG - HTG.TRG) / 2)))
and (CCV <= 0)) then Goto HEAT_ECON_WAIT

Line HEAT_ECON_WAIT
  HTG = Off
  ECON = Off
  MCLG = On
  AHU.MODE.S = "Mechanical Cooling"
  If (not CHWAVAIL) then Goto HEAT_ECON_MODE
  If (OAT > (OAT.HTG.SP - 2)) and ((ST > (HTG.TRG + ((CLG.TRG - HTG.TRG) / 2)))
or (CCV >= 0)) then Goto MECH_MODE
  If (TM >= 10) then Goto HEAT_ECON_MODE
```

```
Line E
  AHU.MODE.E.TIME = Date
  Goto Decision
```

Example:
Based on outside air conditions, this program is used to determine if the system should operate in heating, economizer, or mechanical cooling mode.

HVAC-003 – **STANDARD AIR HANDLER START/STOP PROGRAM**

This logic commands the AHU fan to start, if the relevant interlocks (freezestat, smoke, OOS, power failure) are not in alarm. If any of these interlocks are in alarm, the fan is commanded to stop.

```
Line DECISION
  START.DELAY = Second * 5
  If PowerFail then Goto WAIT_TO_START
  If SFAN then Goto FAN_ON
  Goto FAN_OFF

Line FAN_OFF
  SFAN = Off
  If FREZ or SA.SMOK or AHU.OOS then Goto FAN_OFF
  If (OCC or MWU or MCD or UNCHTG or UNCCLG) and (TS > 30) then Goto
WAIT_TO_START

Line FAN_ON
  SFAN = On
  If PowerFail then Goto WAIT_TO_START
  If FREZ or SA.SMOK or AHU.OOS then Goto FAN_OFF
  If (not OCC) and (not MWU) and (not MCD) and (not UNCHTG) and (not UNCCLG)
and (TS > 300) then Goto FAN_OFF

Line WAIT_TO_START
  SFAN = Off
  If FREZ or SA.SMOK or AHU.OOS then Goto FAN_OFF
  If (not OCC) and (not MWU) and (not MCD) and (not UNCHTG) and (not UNCCLG)
then Goto FAN_OFF
  If (TS > START.DELAY) and PWR.FAIL.STAGE3 then Goto FAN_ON

Line E
  SFAN.E.TIME = Date
  Goto DECISION
```

Example:

Air handling unit supply fan – if the associated freezestat is not in alarm, the supply fan will start. If the freezestat is in alarm, the supply fan will shut down or will not start.

HVAC-004 – STANDARD REHEAT COIL CONTROL VALVE PROGRAM

This is standard logic for the control of a reheat coil (direct or reverse acting).

```
Numeric Kp, Ki, Kd, lerr, db, PIDval, RHCV.ACT, INT.MAX
Datetime LTIME

Line Decision
  RHCV.ACT = 0 ' 1 = DIRECT ACTING.  Set RHCV.ACT to 1 if heating valve is
' a 2-way N.O. valve, or if it is a 3-way valve that is
' open to the hot water coil with 0 psi, 0 volts, or 0 mA
' sent to the valve
' 0 = REVERSE ACTING.  Set RHCV.ACT to 0 if none of the
' above are true.

  Kp = 0.03125
  Ki = 0.0013
  Kd = 0.005625
  db = 0.5
  INT.MAX = 1 / Ki
  lerr = 0
  LTIME = Date
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or MCLG then Goto FULL_CLOSED
  If UNCHTG or (MWU and (ST < HSP)) then Goto FULL_OPEN
  Goto MODULATING

Line FULL_OPEN
  RHCV = 100
  RHCV.INT = (RHCV / 100) / Ki
  LTIME = Date
  lerr = 0
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or MCLG then Goto FULL_CLOSED
  If UNCHTG or (MWU and (ST < HSP)) then Goto FULL_OPEN
  Goto MODULATING

Line FULL_CLOSED
  RHCV = 0
  RHCV.INT = 0
  LTIME = Date
  lerr = 0
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or MCLG then Goto FULL_CLOSED
  If UNCHTG or (MWU and (ST < HSP)) then Goto FULL_OPEN
  Goto MODULATING
```

```
Line MODULATING
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or MCLG then Goto FULL_CLOSED
  If UNCHTG or (MWU and (ST < HSP)) then Goto FULL_OPEN
  If (RHCV State = Disabled) then Goto MODULATING
  If (abs(RHDAT - RHDAT.TRG) > db) and (abs(Date - LTIME) > 2) and (RHCV State =
Enabled) then
    PIDval = PID.F(RHDAT, RHDAT.TRG, Kp, Ki, Kd, 0, RHCV.ACT, Ierr, RHCV.INT,
LTIME)
    RHCV.INT = minimum(RHCV.INT, INT.MAX)
    RHCV = minimum(maximum((PIDval * (RHCV.RHI - RHCV.RLO)) + RHCV.RLO,
RHCV.RLO), RHCV.RHI)
    LTIME = Date
  Else
    If (abs(RHDAT - RHDAT.TRG) <= db) and (abs(Date - LTIME) > 3) then Goto
IN_DEADBAND
    If (RHCV State = Disabled) then
      RHCV.INT = (RHCV / 100) / Ki
      LTIME = Date
    Endif
  Endif

Line IN_DEADBAND
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or MCLG then Goto FULL_CLOSED
  If (abs(RHDAT - RHDAT.TRG) > db) then Goto MODULATING
  LTIME = Date
  Ierr = 0

Line E
  RHCV.E.TIME = Date
  Goto Decision
```

Example:
Reheat coil valve control – as space temperature falls below its active setpoint, the heating valve will modulate open. As the space temperature rises above its active setpoint, the heating valve will modulate closed.

HVAC-005 – STANDARD AIR HANDLER OUTDOOR AIR DAMPER PROGRAM

This is standard logic for control of an outside air damper.

```
Numeric Kp, Ki, Kd, lerr, db, PIDval, OAD.ACT, L1, INT.MAX
Datetime LTIME

Line Decision
  OAD.ACT = 1
  Kp = 0.02
  Ki = 0.0008
  Kd = 0
  db = 1
  INT.MAX = 1 / Ki
  LTIME = Date
  lerr = 0
  If FREZ then Goto FREZ_CLOSED
  If (not SFAN.POR) or SMOK or (not OCC) or MWU or ((MCD or OCC.EST) and (not
ECON)) then Goto FULL_CLOSED
  If (not ECON) and (CO2 < CO2.SP) then Goto MIN_POS
  Goto MODULATING

Line FREZ_CLOSED
  OAD = 0
  RLAD = OAD
  OAD.INT = 0
  LTIME = Date
  lerr = 0
  If (not FREZ) and SFAN.POR then Goto FREZ_OPEN

Line FREZ_OPEN
  If L1 > 30 then
    OAD = OAD + 1
    RLAD = OAD
    OAD.INT = (OAD / 100) / Ki
    L1 = 0
  Endif
  L1 = L1 + 1
  If FREZ then Goto FREZ_CLOSED
  If (OAD > OAD.MIN.POS.SP) then Goto MIN_POS
  LTIME = Date
  lerr = 0

Line FULL_CLOSED
  OAD = 0
  RLAD = OAD
```

```
  OAD.INT = 0
  LTIME = Date
  lerr = 0
  If FREZ or (OAT < 40) then Goto FREZ_CLOSED
  If (not SFAN.POR) or SMOK or (not OCC) or MWU or ((MCD or OCC.EST) and (not
ECON)) then Goto FULL_CLOSED
  If (not ECON) and (CO2 < CO2.SP) then Goto MIN_POS
  Goto MODULATING

Line MIN_POS
  OAD = OAD.MIN.POS.SP
  RLAD = OAD
  OAD.INT = (OAD / 100) / Ki
  LTIME = Date
  lerr = 0
  If FREZ then Goto FREZ_CLOSED
  If (not SFAN.POR) or SMOK or (not OCC) or MWU or ((MCD or OCC.EST) and (not
ECON)) then Goto FULL_CLOSED
  If (not ECON) and (CO2 < CO2.SP) then Goto MIN_POS
  Goto MODULATING

Line MODULATING
  If FREZ then Goto FREZ_CLOSED
  If (not SFAN.POR) or SMOK or (not OCC) or MWU or ((MCD or OCC.EST) and (not
ECON)) then Goto FULL_CLOSED
  If (not ECON) and (CO2 < CO2.SP) then Goto MIN_POS
  If (abs(MAT - MAT.TRG) > db) and (abs(Date - LTIME) > 2) and (OAD State =
Enabled) then
    PIDval = PID.F(MAT, MAT.TRG, Kp, Ki, Kd, 0, OAD.ACT, lerr, OAD.INT, LTIME)
    OAD.INT = minimum(OAD.INT, INT.MAX)
    OAD = minimum(maximum((PIDval * (OAD.RHI - OAD.MIN.POS.SP)) +
OAD.MIN.POS.SP, OAD.MIN.POS.SP), OAD.RHI)
    RLAD = OAD
    LTIME = Date
  Else
    If (abs(MAT - MAT.TRG) <= db) and (abs(Date - LTIME) > 3) then Goto
IN_DEADBAND
    If (OAD State = Disabled) then
      OAD.INT = (OAD / 100) / Ki
      LTIME = Date
      RLAD = OAD
    Endif
  Endif

Line IN_DEADBAND
  If (abs(MAT - MAT.TRG) > db) or (TS > 30) then Goto MODULATING
```

If FREZ then Goto FREZ_CLOSED
If (not SFAN.POR) or SMOK or (not OCC) or MWU or ((MCD or OCC.EST) and (not ECON)) then Goto FULL_CLOSED
If (not ECON) and (CO2 < CO2.SP) then Goto MIN_POS
LTIME = Date
lerr = 0

Line E
OAD.E.TIME = Date
Goto Decision

Example:
Air handling unit – outdoor air damper is fully opened, or opened to minimum outside air position, or modulated based on outside air conditions (economizer controls).

HVAC-006 – STANDARD FACE/BYPASS DAMPER AND PREHEAT VALVE CONTROL

This sequence controls standard face/bypass damper operation and preheat valve control.

```
Numeric Kp, Ki, Kd, lerr, db, PIDval, FBD.ACT, INT.MAX
Datetime LTIME

Line Decision
  FBD.ACT = 0 ' 1 = DIRECT ACTING.  Set FBD.ACT to 1 if heating valve is
' a 2-way N.O. valve, or if it is a 3-way valve that is
' open to the hot water coil with 0 psi, 0 volts, or 0 mA
' sent to the valve
' 0 = REVERSE ACTING.  Set FBD.ACT to 0 if none of the
' above are true.

  Kp = 0.025
  Ki = 0.002
  Kd = 0
  db = 0.25
  INT.MAX = 1 / Ki
  lerr = 0
  LTIME = Date
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or (OAT > 55) or (CCV > 0) then Goto FULL_CLOSED
  Goto MODULATING

Line FULL_OPEN
  PHCV = On
  FBD = 100
  FBD.INT = (FBD / 100) / Ki
  LTIME = Date
  lerr = 0
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or (OAT > 55) or (CCV > 0) then Goto FULL_CLOSED
  Goto MODULATING

Line FULL_CLOSED
  If (OAT < 55) and (CCV <= 0) then
    PHCV = On
    FBD = 100
  Else
    PHCV = Off
    FBD = 0
  Endif
```

```
  FBD.INT = 0
  LTIME = Date
  lerr = 0
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or (OAT > 55) or (CCV > 0) then Goto FULL_CLOSED
  Goto MODULATING

Line MODULATING
  PHCV = On
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or (OAT > 55) or (CCV > 0) then Goto FULL_CLOSED
  If (abs(PHDAT - PHDAT.SP) > db) and (abs(Date - LTIME) > 2) and (FBD State =
Enabled) then
    PIDval = PID.F(PHDAT, PHDAT.SP, Kp, Ki, Kd, 0, FBD.ACT, lerr, FBD.INT, LTIME)
    FBD.INT = minimum(FBD.INT, INT.MAX)
    FBD = minimum(maximum((PIDval * (FBD.RHI - FBD.RLO)) + FBD.RLO, FBD.RLO),
FBD.RHI)
    LTIME = Date
  Else
    If (abs(PHDAT - PHDAT.SP) <= db) and (abs(Date - LTIME) > 3) then Goto
IN_DEADBAND
    If (FBD State = Disabled) then
      FBD.INT = (FBD / 100) / Ki
      LTIME = Date
    Endif
  Endif

Line IN_DEADBAND
  If FREZ then Goto FULL_OPEN
  If (not SFAN.POR) or (OAT > 55) or (CCV > 0) then Goto FULL_CLOSED
  If (abs(PHDAT - PHDAT.SP) > db) then Goto MODULATING
  LTIME = Date
  lerr = 0

Line E
  FBD.E.TIME = Date
  Goto Decision
```

Example:
Air handling unit with integral face and bypass damper – When outside air temperature falls below a fixed setpoint, the heating coil control valve is opened to prevent coil freezing. The face and bypass dampers modulate to maintain a constant supply air temperature.

HVAC-007 - STANDARD EARLY START PROGRAM

This logic determines cool-down and warm-up times for spaces with varying HVAC controls during unoccupied and occupied periods.

```
' IN THE CALC PROGRAM CSP.DLT=(ST-CSP)*(ST>CSP)
' HSP.DLT=(ST-HSP)*(ST<HSP)

Line EST.CALC
  OCC.EST = Off
  MWU = Off
  MCD = Off
  EST.K = average(EST.MINUTES)
  EST.DLT = maximum(abs(HSP.DLT), CSP.DLT)
  EST = (minimum((maximum((EST.DLT * EST.K), 15)), 180))
  If OCC then Goto OCCUPIED
  Goto DECISION

Line OCCUPIED
  OCC.EST = Off
  MWU = Off
  MCD = Off
  If (not OCC) then Goto EST.CALC

Line DECISION
  OCC.EST = Off
  MWU = Off
  MCD = Off
  If (Date > ((START.TIME.L) - (EST * 60))) and (Date <= START.TIME.L) then Goto
RECORDTIME
  Goto EST.CALC

Line RECORDTIME
  OCC.EST = Off
  MWU = Off
  MCD = Off
  EST.DLT = maximum(abs(HSP.DLT), CSP.DLT)
  EST.TIME = Date
  If (CSP.DLT > abs(HSP.DLT)) and MCLG and (abs(CSP.DLT) > 1) then Goto
COOLDOWN
  If (abs(HSP.DLT) > 1) then Goto WARMUP
  If OCC.CYBER then Goto OCCUPIED

Line COOLDOWN
  OCC.EST = Off
```

```
  MWU = Off
  MCD = not OCC.CYBER
  If (CSP.DLT <= 1) then Goto CALCULATECOOL

Line WARMUP
  OCC.EST = Off
  MWU = not OCC.CYBER
  MCD = Off
  If (HSP.DLT > -1) then Goto CALCULATEWARM

Line CALCULATECOOL
  CSP.DLT.OCC = CSP.DLT
  EST.MINUTES = ((Date - EST.TIME) / 60) / EST.DLT
  If not OCC.CYBER then Goto COOLWAIT
  Goto OCCUPIED
Line COOLWAIT
  OCC.EST = On
  MWU = Off
  MCD = Off
  If OCC.CYBER then Goto OCCUPIED

Line WARMWAIT
  OCC.EST = On
  MWU = Off
  MCD = Off
  If OCC.CYBER then Goto OCCUPIED

Line CALCULATEWARM
  HSP.DLT.OCC = HSP.DLT
  EST.MINUTES = ((Date - EST.TIME) / 60) / EST.DLT
  If not OCC.CYBER then Goto WARMWAIT
  Goto OCCUPIED

Line E
  EST.E.TIME = Date
  Goto DECISION
```

Example:
This program determines HVAC system start times for morning warm-up and cool-down operating modes based on outside air conditions.

HVAC-008 – STANDARD TARGET RESET PROGRAM

This control function sets the discharge air temperature set point for a target reset control scheme. Target reset control varies the discharge temperature, depending on the space temperature to achieve a target space temperature.

Required External Software Numeric Points:
TARGET.rhi, TARGET.rlo, TARGET.PID

Program Definition:
Numeric actn, CO, TARGETkp, TARGETki, TARGETkd, TARGETdbnd, TARGETlerr

```
INITIALIZE:
  actn = -1
  TARGETkp = 0.1
  TARGETki = 0.01
  TARGETkd = 0.001
  TARGETdbnd = 0.2 'Deg F
  TARGET.rhi = 75
  TARGET.rlo = 55
  Goto CNTRL

CNTRL:
  If ((abs(CONTROL.POINT - CONTROL.POINT.SETPOINT)) < TARGETdbnd) then
Goto IN.DeadBand
  PID.F(CONTROL.POINT, CONTROL.POINT.SETPOINT, TARGETkp, TARGETki,
TARGETkd, actn, TARGETlerr, TARGETint, CO)
  TARGET.PID = CO
  TARGET = ((TARGET.PID * (TARGET.rhi - TARGET.rlo)) + TARGET.rlo)
  Goto MODU.WAIT

MODU.WAIT:
  If (TS >= 2) then Goto CNTRL

IN.DeadBand:
  If ((abs(CONTROL.POINT - CONTROL.POINT.SP)) > TARGETdbnd) or (TS > 60)
then Goto CNTRL
```

Variable Definitions:
actn = action (1=DA/-1=RA)
kp = proportional gain
ki = integral gain
kd = derivative
dbnd = deadband
lerr = error

int = integral value
rhi = control range high value
rlo = control range low value
CO = PID function returned value
TARGET = controlled variable
CONTROL.POINT = reference variable
CONTROL.POINT.SETPOINT = reference variable setpoint value
PID.F = PID function call

Example:
Discharge air temperature reset, based on measured space temperature relative to its scheduled setpoint value. As the space temperature falls below its active setpoint, the discharge air target will increase. As the space temperature rises above its active setpoint, the discharge air target will decrease.

HVAC-009 – **STANDARD MISCELANEOUS CALCULATIONS PROGRAM**

This program controls the heating and cooling control for HVAC systems.

```
Numeric csp.old, hsp.old, PH1, PH2, KI, K2
Numeric SLIDER.VMAX, SLIDER.VMID, SLIDER.VMIN

Line DECISION
  SLIDER.VMAX = 2.64
  SLIDER.VMID = 1.8
  SLIDER.VMIN = 0.1
  KI = 0.02
  K2 = 0.02
  Goto 1

Line 1
  DEHUMID.ADJ = LINEFIT.F(RARH, 50, 65, 0, 25)
  If RARH > 60 and CHWAVAIL and HWAVAIL and SFAN.POR then Set DEHUMID =
On
  If RARH < 55 or not CHWAVAIL or not HWAVAIL or not SFAN.POR then Set
DEHUMID = Off
  OCC = maximum(OCC.CYBER, OVR, OCC.EST)
  If (CSP <> csp.old) and ((CSP - HSP) < 2) then
    HSP and hsp.old = CSP - 2
    csp.old = CSP
  Endif
  If (HSP <> hsp.old) and ((CSP - HSP) < 2) then
    CSP and csp.old = HSP + 2
    hsp.old = HSP
  Endif
  If (ST.ADJ.RAW >= SLIDER.VMID) then ST.ADJ = LINEFIT.F(ST.ADJ.RAW,
SLIDER.VMID, SLIDER.VMAX, 0, ST.ADJ.SP)
  If (ST.ADJ.RAW < SLIDER.VMID) then ST.ADJ = LINEFIT.F(ST.ADJ.RAW,
SLIDER.VMID, SLIDER.VMIN, 0, -ST.ADJ.SP)
  ST.FLTR = (((ST.FLTR * 0.96) + (ST * 0.04)) * (ST < 99 & (ST > -99))) + (ST.FLTR *
(ST >= 99 or (ST <= -99)))
  HSP.DLT = (ST.FLTR - (HSP + ST.ADJ)) * (ST.FLTR < (HSP + ST.ADJ))
  CSP.DLT = (ST.FLTR - (CSP + ST.ADJ)) * (ST.FLTR > (CSP + ST.ADJ))
  CLG.TRG = ((CSP + DTAJ - MCD + ST.ADJ) * (OCC or MCD)) + (UNCSP * ((not
OCC) and (not MCD)))
  HTG.TRG = ((HSP - DTAJ + MWU + ST.ADJ) * (OCC or MWU)) + (UNHSP * ((not
OCC) and (not MWU)))
  ST.HDLT = ST.FLTR - HTG.TRG
  ST.CDLT = ST.FLTR - CLG.TRG
  If OCC then
    CCLG and PH1 = (ST.FLTR > (CLG.TRG + 1))
```

```
    CHTG and PH2 = (ST.FLTR < HTG.TRG) or (RHCV > 0)
  Else
    If ST.FLTR > UNCSP then CCLG = On
    If CCLG = On and (ST.FLTR < (UNCSP - 4) or (PH1 and (ST.FLTR < UNCSP))) then
CCLG and PH1 = Off
    If ST.FLTR < UNHSP then CHTG = On
    If CHTG = On and (ST.FLTR > (UNHSP + 4) or (PH2 and (ST.FLTR > UNHSP)))
then CHTG and PH2 = Off
  Endif
  UNCHTG = (not OCC) and CHTG
  UNCCLG = (not OCC) and CCLG
  MAT.TRG = DAT.TRG - 2
  DA.SPR.FLTR = FLTR.F(DA.SPR, DA.SPR.FLTR, 1)
  OAD.MIN.POS.TRG = minimum(maximum(((OAD.MIN.POS.TRG + ((CO2 -
CO2.HI.SP) * KI))), OAD.MIN.POS.SP), 100)
  SFAN.VSD.MAX = minimum(maximum(((SFAN.VSD.MAX) - ((DA.SPR.FLTR -
(DA.SPR.HI.SP - 0.5)) * K2)), SFAN.VSD.MIN), 100)
  Goto 2

Line 2
  SFAN.HIST = SFAN
  SFAN.POR = (SFAN.VSD.FDB > 20)
  RFAN.HIST = RFAN
  RFAN.POR = (RFAN.VSD.FDB > 20)
  SSPR.FLTR = FLTR.F(SSPR, SSPR.FLTR, 1)
  DA.SPR.FLTR = FLTR.F(DA.SPR, DA.SPR.FLTR, 1)
  LGTS.KW = ((LGTS.VOLTS * LGTS.AMPS * LGTS.PF) / 1000)
'  TWO-PHASE   LGTS.KW = ((LGTS.VOLTS * LGTS.AMPS * LGTS.PF * 2) / 1000)
'  THREE-PHASE   LGTS.KW = ((LGTS.VOLTS * LGTS.AMPS * LGTS.PF * 1.73) /
1000)
  Goto 1

Line E
  CALC.E.TIME = Date
  Goto 1
```

Example:
This program controls HVAC systems in heating, cooling, or dehumidification mode based on space temperature and/or space humidity conditions.

HVAC-010 – STANDARD OCCUPANCY OVERRIDE PROGRAM

This sequence allows the occupancy setting to be overridden by the space temperature.

```
' TRIGGERED BY SPACE TEMPERATURE

Line DECISION
  If (ST > 120) then Goto OVERRIDE_ON
  Goto OVERRIDE_OFF

Line OVERRIDE_ON
  OVR = On
  If (ST > 120) then Goto RESET_TIMER
  If (TM < OVR.TM) then Goto OVERRIDE_ON
  Goto OVERRIDE_OFF

Line RESET_TIMER
  If (ST > 120) then Goto RESET_TIMER
  Goto OVERRIDE_ON

Line OVERRIDE_OFF
  OVR = Off
  Stop
```

Example:
Air handling unit – if the unit is in unoccupied mode of operation, and the space temperature falls below setpoint, the unit will be switched to occupied mode until the space temperature reaches the setpoint.

HVAC-011 – STANDARD PULSE WIDTH VALVE CONTROL PROGRAM

This is standard pulse width control for a heating control valve.

```
Numeric KP, DB, HCV.MAXP, HCV.PULSE

Line DECISION
  HCV.MAXP = 150
  KP = 5.55
  DB = 0.5
  If FREZ or WTR.BAL or ((HCV.POS State = Disabled) and (HCV.POS = 100)) then
Goto FULL_OPEN
  If ((HCV.POS State = Disabled) and (HCV.POS = 0)) or DX then Goto FULL_CLOSED
  If (UNCHTG or MWU) and (ST < HSP) then Goto FULL_OPEN
  If (not SFAN.POR) or (not HWAVAIL) or (MCD and (ST > CSP)) then Goto
FULL_CLOSED
  If ((abs(DAT - DAT.TRG)) < DB) then Goto IN_DEADBAND
  Goto MODULATING

Line IN_DEADBAND
  HCV = Off
  If FREZ or WTR.BAL or ((HCV.POS State = Disabled) and (HCV.POS = 100)) then
Goto FULL_OPEN
  If ((HCV.POS State = Disabled) and (HCV.POS = 0)) or DX then Goto FULL_CLOSED
  If (UNCHTG or MWU) and (ST < HSP) then Goto FULL_OPEN
  If (not SFAN.POR) or (not HWAVAIL) or (MCD and (ST > CSP)) then Goto
FULL_CLOSED
  If ((abs(DAT - DAT.TRG)) < DB) then Goto IN_DEADBAND
  Goto MODULATING

Line FULL_OPEN
  HCV = HCV.MAXP
  HCV.POS = 100
  If FREZ or ((HCV.POS State = Disabled) and (HCV.POS = 100)) then Goto
FULL_OPEN
  If ((HCV.POS State = Disabled) and (HCV.POS = 0)) then Goto RESET_OUTPUT
  If (TS > HCV.MAXP) and ((not UNCHTG and not MWU) or (ST > HSP) or (not
HWAVAIL)) and (not WTR.BAL) then Goto RESET_OUTPUT

Line FULL_CLOSED
  HCV = -HCV.MAXP
  HCV.POS = 0
  If FREZ then Goto RESET_OUTPUT
  If ((HCV.POS State = Disabled) and (HCV.POS = 0)) or DX then Goto FULL_CLOSED
  If ((UNCHTG or MWU) and (ST < HSP)) or WTR.BAL or ((HCV.POS State = Disabled)
and (HCV.POS = 100)) then Goto RESET_OUTPUT
```

```
  If SFAN.POR and HWAVAIL and (TS > HCV.MAXP) and ((not MCD) or (ST < CSP))
then Goto RESET_OUTPUT

Line RESET_OUTPUT
  HCV = Off
  Goto DECISION

Line MODULATING
  If ((abs(DAT - DAT.TRG)) > DB) then
    If DAT - DAT.TRG > 0 then
      HCV = minimum(-0.1, (maximum(((DAT.TRG - DAT) * KP), -HCV.MAXP)))
      HCV.PULSE = minimum(-0.1, (maximum(((DAT.TRG - DAT) * KP), -HCV.MAXP)))
      HCV.POS = minimum((maximum(((HCV / HCV.MAXP) * 100 + HCV.POS), 0)), 100)
    Else
      HCV = maximum(0.1, (minimum(((DAT.TRG - DAT) * KP), HCV.MAXP)))
      HCV.PULSE = maximum(0.1, (minimum(((DAT.TRG - DAT) * KP), HCV.MAXP)))
      HCV.POS = minimum((maximum(((HCV / HCV.MAXP) * 100 + HCV.POS), 0)), 100)
    Endif
 Endif
  Goto WAIT_1

Line WAIT_1
  If FREZ then Goto RESET_OUTPUT
  If not SFAN.POR or (TS > 150) then Goto RESET_OUTPUT
```

Example:
This program will open, close, or modulate a heating coil control valve as required to maintain setpoint.

HP-001 – STANDARD DX COOLING CONTROL PROGRAM

This sequence is for the control of cooling via a direct expansion heat pump.

```
Line DECISION
  If DX then Goto DX_ON
  Goto DX_OFF

Line DX_OFF
  DX = Off
  If (not SFAN.POR) or (not MCLG) or (DAT < DAT.LOW.SP) then Goto DX_OFF
  If (TS < 240) then Goto DX_OFF
  If ((ST.CDLT > 1) and OCC) or MCD or UNCCLG then Goto DX_ON

Line DX_ON
  DX = On
  If (not SFAN.POR) or (not MCLG) or (DAT < DAT.LOW.SP) then Goto DX_OFF
  If (TS < 240) then Goto DX_ON
  If (ST.CDLT < -1) and (OCC or MCD) then Goto DX_OFF
  If (not OCC) and (not MCD) and (not UNCCLG) then Goto DX_OFF

Line E
  Goto DECISION
```

Example:

This program is used to control equipment such as packaged heat pump units as required to maintain space cooling setpoint.

HP-002 – STANDARD FREEZE PROTECTION PUMP PROGRAM

This program is for standard freeze protection for a heat pump.

```
Line DECISION
  If FREZ.PUMP then Goto PUMP_ON
  Goto PUMP_OFF

Line PUMP_OFF
  FREZ.PUMP = Off
  If FREZ or (OAT < OA.FREZ.PUMP.SP) then Goto PUMP_ON

Line PUMP_ON
  FREZ.PUMP = On
  If (not FREZ) and (OAT > (OA.FREZ.PUMP.SP + 2)) and (TS > 300) then Goto
PUMP_OFF

Line E
  FREZ.PUMP.E.TIME = Date
  Goto DECISION
```

Example:

Air handling unit supplying 100% outside air – if the outside air temperature falls below setpoint, the freeze protection pump will be started to prevent coil freezing. When the outside air temperature rises about setpoint, the pump will be turned off.

HP-003 – STANDARD SMART SENSOR DISPLAY PROGRAM

This logic is to set up space temperature and adjustment for a smart sensor display.

```
Line DISPLAY_TEMP
  LCDdisplay[2] = 11
  LCDdisplay[1] = ST
  LCDdisplay[7] = 1024
  If (LCDdisplay[0] = 2) or (LCDdisplay[0] = 5) then Goto ADJUST_SP

Line ADJUST_SP
  LCDdisplay[2] = 11
  LCDdisplay[1] = (ST.TRG + ST.ADJ)
  LCDdisplay[7] = (1024 + 2048)
  If LCDdisplay[0] = 2 then
    ST.ADJ = minimum(maximum((ST.ADJ + 0.5), -ST.ADJ.SP), ST.ADJ.SP)
    Goto ADJUST_SP
  Endif
  If LCDdisplay[0] = 5 then
    ST.ADJ = minimum(maximum((ST.ADJ - 0.5), -ST.ADJ.SP), ST.ADJ.SP)
    Goto ADJUST_SP
  Endif
  If (LCDdisplay[0] = 6) or (TS > 30) then Goto DISPLAY_TEMP
```

EXAMPLE:

This routine will adjust the smart sensor display in a temperature measurement application.

PUMP-001 – STANDARD PUMP START/STOP PROGRAM

This is standard logic for controlling multiple pumps to be on or off in various configurations.

```
Line DECISION
  If CHWP5 and CHWP6 then Goto BOTH_ON
  If CHWP5 then Goto CHWP5_ON
  If CHWP6 then Goto CHWP6_ON
  Goto CHWP_OFF

Line CHWP_OFF
  CHWP5 = Off
  CHWP6 = Off
  If (not CHWSYS.RUN) or (TS < 30) then Goto CHWP_OFF
  If CHWP5.LEAD then Goto CHWP5_ON
  Goto CHWP6_ON

Line CHWP5_ON
  CHWP5 = On
  CHWP6 = Off
  If (not CHWSYS.RUN) and (TM > SCHWP.MIN.ON) and (NO.CHWP.ON <= 0) then
Goto CHWP_OFF
  If (not CHWP5.POR) then Goto CHWP5_WAIT
  If (not CHWP5.LEAD) then Goto BOTH_ON
  If SCHWP.LAG then Goto BOTH_ON

Line CHWP5_WAIT
  CHWP5 = On
  CHWP6 = Off
  If CHWP5.POR then Goto CHWP5_ON
  If (TS > 60) then Goto BOTH_ON

Line BOTH_ON
  CHWP5 = On
  CHWP6 = On
  If (TS < 5) then Goto BOTH_ON
  If (not CHWSYS.RUN) and (TM > SCHWP.MIN.ON) and (NO.CHWP.ON <= 0) then
Goto CHWP_OFF
  If SCHWP.LAG then Goto BOTH_ON
  If CHWP5.LEAD and CHWP5.POR then Goto CHWP5_ON
  If (not CHWP5.LEAD) and CHWP6.POR then Goto CHWP6_ON

Line CHWP6_ON
  CHWP5 = Off
  CHWP6 = On
```

```
  If (not CHWSYS.RUN) and (TM > SCHWP.MIN.ON) and (NO.CHWP.ON <= 0) then
Goto CHWP_OFF
  If (not CHWP6.POR) then Goto CHWP6_WAIT
  If CHWP5.LEAD then Goto BOTH_ON
  If SCHWP.LAG then Goto BOTH_ON

Line CHWP6_WAIT
  CHWP5 = Off
  CHWP6 = On
  If CHWP6.POR then Goto CHWP6_ON
  If (TS > 60) then Goto BOTH_ON

Line E
  CHWP5_6.E.TIME = Date
  Goto DECISION
```

Example:
This program is used to control operation of two pumps. Based on input, such as from system operator, either pump will run, both pumps will run, or no pumps will run.

PUMP-002 – STANDARD LAG PUMP CONTROL PROGRAM

This is a standard routine to operate a lag pump in a lead/lag configuration.

```
Line DECISION
  If SCHWP.LAG then Goto SCHWP.LAG.ON
  Goto SCHWP.LAG.OFF

Line SCHWP.LAG.OFF
  SCHWP.LAG = Off
  If SCHWP.POR and ((CHW.DPR.SP - CHW.DPR.FLTR) > 2) and SCHWP.MAX.VSD
>= 100 then Goto LAG.DELAY1

Line LAG.DELAY1
  SCHWP.LAG = Off
  If (not SCHWP.POR) or ((CHW.DPR.SP - CHW.DPR.FLTR) < 2) or
(SCHWP.MAX.VSD < 100) then Goto SCHWP.LAG.OFF
  If (TM > LAG.DELAY) then Goto SCHWP.LAG.ON

Line SCHWP.LAG.ON
  SCHWP.LAG = On
  If ((SCHWP.MAX.VSD < 50) and (TM > 10)) then Goto SCHWP.LAG.OFF

Line E
  SCHWP.LAG.E.TIME = Date
  Goto DECISION
```

Example:

Operating and stand-by heating water pumps – if the lead pump fails, the lag pump will be started.

CHILLER-001 – STANDARD CHILLER STAGING PROGRAM

This is a standard routine for starting multiple chillers connected as a system.

```
Line DECISION
  If CLR.STAGE3 then Goto CLR.STAGE3.ON
  If CLR.STAGE2 then Goto CLR.STAGE2.ON
  If CLR.STAGE1 then Goto CLR.STAGE1.ON
  Goto CLR.STAGES.OFF

Line CLR.STAGES.OFF
  CLR.STAGE1 = Off
  CLR.STAGE2 = Off
  CLR.STAGE3 = Off
  CLR.STAGE4 = Off
  If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
  If CHWSYS.RUN and SCHWP.POR and (TM > CLR.MIN.OFF) then Goto
CLR.DELAY1

Line CLR.DELAY1
  CLR.STAGE1 = Off
  CLR.STAGE2 = Off
  CLR.STAGE3 = Off
  CLR.STAGE4 = Off
  If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
  If (not CHWSYS.RUN) or (not SCHWP.POR) then Goto CLR.STAGES.OFF
  If (TM > CLR.START.DELAY) then Goto CLR.STAGE1.ON

Line CLR.STAGE1.ON
  CLR.STAGE1 = On
  CLR.STAGE2 = Off
  CLR.STAGE3 = Off
  CLR.STAGE4 = Off
  If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
  If ((not CHWSYS.RUN) or (not SCHWP.POR)) and (TM > CLR.MIN.ON) then Goto
CLR.STAGES.OFF
  If CHWS < CHWS.STAGEUP.SP and not CHWP7.FAIL and not COMP1_2.ALM then
Goto DECISION
  If ((CHWS > CHWS.STAGEUP.SP) and (TM > 30)) or (CHWP7.FAIL and (not
CHWP8.FAIL)) or COMP1_2.ALM and STAGE3.LOADSHED = Off then Goto
CLR.DELAY2

Line CLR.DELAY2
  CLR.STAGE1 = On
  CLR.STAGE2 = Off
  CLR.STAGE3 = Off
```

CLR.STAGE4 = Off
If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
If ((not CHWSYS.RUN) or (not SCHWP.POR)) and (TM > CLR.STAGE.DELAY) then Goto CLR.STAGES.OFF
If (TM > CLR.STAGE.DELAY) or CHWP7.FAIL or COMP1_2.ALM then Goto CLR.STAGE2.ON

Line CLR.STAGE2.ON
CLR.STAGE1 = Off
CLR.STAGE2 = On
CLR.STAGE3 = Off
CLR.STAGE4 = Off
If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
If ((not CHWSYS.RUN) or (not SCHWP.POR)) and (TM > CLR.MIN.ON) then Goto CLR.STAGES.OFF
If ((TM > 30) and (CHWS > CHWS.STAGEUP.SP)) or (CHWP8.FAIL and (not CHWP7.FAIL)) then Goto CLR.DELAY3
If ((TM > 30) and (CHWS < CHWS.STAGEDN.SP) and (TM > CLR.MIN.ON) and (not CHWP7.FAIL) and (not COMP1_2.ALM)) or STAGE3.LOADSHED = On then Goto CLR.STAGE1.ON

Line CLR.DELAY3
CLR.STAGE1 = Off
CLR.STAGE2 = On
CLR.STAGE3 = Off
CLR.STAGE4 = Off
If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
If ((not CHWSYS.RUN) and (TM > CLR.STAGE.DELAY)) or (not SCHWP.POR) then Goto CLR.STAGES.OFF
If (TM > CLR.STAGE.DELAY) or CHWP8.FAIL then Goto CLR.STAGE3.ON

Line CLR.STAGE3.ON
CLR.STAGE1 = Off
CLR.STAGE2 = Off
CLR.STAGE3 = On
CLR.STAGE4 = Off
If CHWP7.FAIL and CHWP8.FAIL then Goto CLR.STAGES.OFF
If ((not CHWSYS.RUN) and (TM > CLR.MIN.ON)) or (not SCHWP.POR) then Goto CLR.STAGES.OFF
If ((CLR1.MAX.CMP.SPD <= CLR1.MIN.SPD.SP) and (CHWS < CHWS.STAGEDN.SP) and (TM > CLR.MIN.ON)) or (CHWP7.FAIL and (not CHWP8.FAIL)) then Goto CLR.STAGE2.ON

Line E
CLR.STG.E.TIME = Date

Goto DECISION

Example:
This program is used to start and stop individual chillers which are part of a multiple chiller installation, as required to maintain chilled water supply temperature setpoint.

CHILLER-002 – MECHANICAL COOLING DECISION PROGRAM

This routine is decision logic which determines when to run a chilled water system, based on outside air temperature.

```
Line DECISION
  If CHWSYS.RUN then Goto CHWSYS_RUN
  Goto CHWSYS_OFF

Line CHWSYS_OFF
  CHWSYS.RUN = Off
  If CLR.RUN and ((OAT > OA.CLG.SP) or (AHU.CCLG > 5)) and (TM >
CHWSYS.MIN.OFF) then Goto CHWSYS_RUN

Line CHWSYS_RUN
  CHWSYS.RUN = On
  If ((not CLR.RUN) or ((OAT < (OA.CLG.SP - 2)) and (AHU.CCLG < 3))) and (TM >
CHWSYS.MIN.ON) then Goto CHWSYS_OFF

Line E
  CHWSYS.E.TIME = Date
  Goto DECISION
```

Example:

This program is used to start a chilled water system - chillers, pumps, and cooling tower - based on outside air temperature. If the outside air temperature rises above setpoint, the system is started. When the outside air temperature falls below the setpoint, the system is turned off.

TOWER-001 – STANDARD COOLING TOWER FAN LEAD/LAG CONTROL PROGRAM

This is standard routine for operating cooling fans in a lead/lag configuration, where the lead is alternated automatically on a regular basis.

```
'Tri-M Standard LEADLAG01

Line DECISION
  If CT1.FAN1.LEAD then Goto CT1.FAN1LEAD
  Goto CT1.FAN2LEAD

Line CT1.FAN1LEAD
  CT1.FAN1.LEAD = On
  CT1.FAN2.LEAD = Off
  If (Weekday = Wednesday) and (Hourofday = 8) then Goto SWITCHTO2

Line SWITCHTO2
  CT1.FAN1.LEAD = Off
  CT1.FAN2.LEAD = On
  If (Weekday <> Wednesday) or (Hourofday >= 9) then Goto CT1.FAN2LEAD

Line CT1.FAN2LEAD
  CT1.FAN1.LEAD = Off
  CT1.FAN2.LEAD = On
  If (Weekday = Wednesday) and (Hourofday = 8) then Goto SWITCHTO1

Line SWITCHTO1
  CT1.FAN1.LEAD = On
  CT1.FAN2.LEAD = Off
  If (Weekday <> Wednesday) or (Hourofday >= 9) then Goto CT1.FAN1LEAD

Line E
  LL.CT1.E.TIME = Date
  Goto DECISION
```

Example:
When a cooling tower includes two fans, this program is used to automatically alternate the lead fan and the lag fan at a specific time on a specific day of the week.