1

124 lines - 6 Removals

```
"""An eclss_pressure model.

variable :: TypeOfVariable

(variable) = optional variable

                          _____
clock                     :: Clock
 =>  |                        | ->  en_needed_pres  ::
 State
design                    :: DomeSpec
 =>  |                        |
(available energy)        :: Energy
 =>  |   EclssPressureEnv     |
(strucure_health)         :: Struct
 =>  |                        |
(pressure setpoints)      :: HM
 =>  |                        | ->  en_used_pres    ::
 State
(int_pressure)            :: InteriorEnv
 =>  _________________________


"""


__all__ = ["make_eclss_pressure_env_0"]


from cdcm import *

def make_eclss_pressure_env_0(dome_specs,
                              energy_available_energy,
                              struct_health,
                              int_env_pres,
                              HM_pressure_lower_setpoin
t,
                              HM_pressure_upper_setpoin
t):
    # print('zz',energy_available_energy,struct_health,
int_env_pres,HM_pressure_lower_setpoint,HM_pressure_upp
er_setpoint)
    with System(name="eclss_pressure", description="The
eclss_pressure environment") as eclss_pressure:
        en_needed_pres = State(name="en_needed_pres",
                               value=0.0,
                               units="J",
                               description="Energy
needed to control pressure")
        en_used_pres = State(name="en_used_pres",
                             value=0.0,
                             units="J",
                             description="Energy u
```

124 lines + 6 Additions

```
"""An eclss_pressure model.

variable :: TypeOfVariable

(variable) = optional variable

                          _____
clock                     :: Clock
 =>  |                        | ->  en_needed_pres  ::
 State
design                    :: DomeSpec
 =>  |                        |
(available energy)        :: Energy
 =>  |   EclssPressureEnv     |
(strucure_health)         :: Struct
 =>  |                        |
(pressure setpoints)      :: HM
 =>  |                        | ->  en_used_pres    ::
 State
(int_pressure)            :: InteriorEnv
 =>  _________________________


"""


__all__ = ["make_eclss_pressure_env_0"]


from cdcm import *

def make_eclss_pressure_env_0(dome_specs,
                              energy_available_energy,
                              struct_health,
                              int_env_pres,
                              HM_pressure_lower_setpoin
t,
                              HM_pressure_upper_setpoin
t):
    # print('zz',energy_available_energy,struct_health,
int_env_pres,HM_pressure_lower_setpoint,HM_pressure_upp
er_setpoint)
    with System(name="eclss_pressure", description="The
eclss_pressure environment") as eclss_pressure:
        en_needed_pres = State(name="en_needed_pres",
                               value=0.0,
                               units="J",
                               description="Energy
needed to control pressure")
        en_used_pres = State(name="en_used_pres",
                             value=0.0,
                             units="J",
                             description="Energy u
```

```
sed to control pressure")
        # structure_sec_1 = Variable(name="structure_sec_1",
        #                            value=struct_health.value[0],
        #                            units="",
        #                            description="health level of the dome section 1; 1 is the healthiest")
        # structure_sec_2 = Variable(name="structure_sec_2",
        #                            value=struct_health.value[1],
        #                            units="",
        #                            description="health level of the dome section 2; 1 is the healthiest")
        # structure_sec_3 = Variable(name="structure_sec_3",
        #                            value=struct_health.value[2],
        #                            units="",
        #                            description="health level of the dome section 3; 1 is the healthiest")
        # structure_sec_4 = Variable(name="structure_sec_4",
        #                            value=struct_health.value[3],
        #                            units="",
        #                            description="health level of the dome section 4; 1 is the healthiest")
        # structure_sec_5 = Variable(name="structure_sec_5",
        #                            value=struct_health.value[4],
        #                            units="",
        #                            description="health level of the dome section 5; 1 is the healthiest")
        # print(struct_health)
        @make_function(en_needed_pres,
                       en_used_pres)
        def f_eclss_pres(pres_capac_per_vol=dome_specs.pres_capac_per_vol,
                                            air_leak_coeficent=dome_specs.air_leak_coeficent,
                         efficiency_of_PM=dome_specs.efficiency_of_PM,
                         lower_pressure_setpo=HM_pressure_lower_setpoint,
                         upper_pressure_setpo=HM_pressure_upper_setpoint,
                         en_needed_pres=en_needed_pres,
                         available_en=energy_available_energy,
                         struct_health=struct_health,
                         int_env_pres=int_env_pres,
```

```
sed to control pressure")
        # structure_sec_1 = Variable(name="structure_sec_1",
        #                            value=struct_health.value[0],
        #                            units="",
        #                            description="health level of the dome section 1; 1 is the healthiest")
        # structure_sec_2 = Variable(name="structure_sec_2",
        #                            value=struct_health.value[1],
        #                            units="",
        #                            description="health level of the dome section 2; 1 is the healthiest")
        # structure_sec_3 = Variable(name="structure_sec_3",
        #                            value=struct_health.value[2],
        #                            units="",
        #                            description="health level of the dome section 3; 1 is the healthiest")
        # structure_sec_4 = Variable(name="structure_sec_4",
        #                            value=struct_health.value[3],
        #                            units="",
        #                            description="health level of the dome section 4; 1 is the healthiest")
        # structure_sec_5 = Variable(name="structure_sec_5",
        #                            value=struct_health.value[4],
        #                            units="",
        #                            description="health level of the dome section 5; 1 is the healthiest")
        # print(struct_health)
        @make_function(en_needed_pres,
                       en_used_pres)
        def f_eclss_pres(en_needed_pres=en_needed_pres,
                         available_en=energy_available_energy,
                         struct_health=struct_health,
                         int_env_pres=int_env_pres,
                         pres_capac_per_vol=dome_specs.pres_capac_per_vol,
                                            air_leak_coeficent=dome_specs.air_leak_coeficent,
                         efficiency_of_PM=dome_specs.efficiency_of_PM,
                         lower_pressure_setpo=HM_pressure_lower_setpoint,
                         upper_pressure_setpo=HM_pressu
```

```
                        ):
        # def f_eclss_pres(lower_pressure_setpo=HM_pressure_lower_setpoint,
        #                  upper_pressure_setpo=HM_pressure_upper_setpoint,
        #                  pres_capac_per_vol=dome_specs.pres_capac_per_vol,
        #                  air_leak_coeficent=dome_specs.air_leak_coeficent,
        #                  efficiency_of_PM=dome_specs.efficiency_of_PM,
        #                  en_needed_pres=en_needed_pres,
        #                  available_en=energy_available_energy,
        #                  structure_sec_1=structure_sec_1,
        #                  structure_sec_2=structure_sec_2,
        #                  structure_sec_3=structure_sec_3,
        #                  structure_sec_4=structure_sec_4,
        #                  structure_sec_5=structure_sec_5,
        #                  int_env_pres=int_env_pres):

            """Transition function for ECLSS heat"""
            print('eclss', en_needed_pres,
                  available_en,
                  struct_health,
                  int_env_pres,
                  lower_pressure_setpo,
                  upper_pressure_setpo,
                  pres_capac_per_vol,
                  air_leak_coeficent,
                  efficiency_of_PM)
            raise Exception('stop_AMIR')
            sadsasa
            # input('sds')
            structure_sec_1 = struct_health[0]
            structure_sec_2 = struct_health[1]
            structure_sec_3 = struct_health[2]
            structure_sec_4 = struct_health[3]
            structure_sec_5 = struct_health[4]
            # print(structure_sec_1, structure_sec_2, structure_sec_3, structure_sec_4, structure_sec_5)
            latenet_structure_eclss = (structure_sec_1 +
                                       structure_sec_2 +
                                       structure_sec_3 +
                                       structure_sec_4 +
                                       structure_sec_5) / 5
            en_needed_pres_new = max(0.0, ((lower_pressure_setpo +
                                            upper_pressure_setpo) / 2 -
                                           int_env_pr
```

```
re_upper_setpoint
                        ):
        # def f_eclss_pres(lower_pressure_setpo=HM_pressure_lower_setpoint,
        #                  upper_pressure_setpo=HM_pressure_upper_setpoint,
        #                  pres_capac_per_vol=dome_specs.pres_capac_per_vol,
        #                  air_leak_coeficent=dome_specs.air_leak_coeficent,
        #                  efficiency_of_PM=dome_specs.efficiency_of_PM,
        #                  en_needed_pres=en_needed_pres,
        #                  available_en=energy_available_energy,
        #                  structure_sec_1=structure_sec_1,
        #                  structure_sec_2=structure_sec_2,
        #                  structure_sec_3=structure_sec_3,
        #                  structure_sec_4=structure_sec_4,
        #                  structure_sec_5=structure_sec_5,
        #                  int_env_pres=int_env_pres):

            """Transition function for ECLSS heat"""
            print('eclss', en_needed_pres,
                  available_en,
                  struct_health,
                  int_env_pres,
                  lower_pressure_setpo,
                  upper_pressure_setpo,
                  pres_capac_per_vol,
                  air_leak_coeficent,
                  efficiency_of_PM)
            raise Exception('stop_AMIR')
            sadsasa
            # input('sds')
            structure_sec_1 = struct_health[0]
            structure_sec_2 = struct_health[1]
            structure_sec_3 = struct_health[2]
            structure_sec_4 = struct_health[3]
            structure_sec_5 = struct_health[4]
            # print(structure_sec_1, structure_sec_2, structure_sec_3, structure_sec_4, structure_sec_5)
            latenet_structure_eclss = (structure_sec_1 +
                                       structure_sec_2 +
                                       structure_sec_3 +
                                       structure_sec_4 +
                                       structure_sec_5) / 5
            en_needed_pres_new = max(0.0, ((lower_pressure_setpo +
                                            upper_pressure_setpo) / 2 -
                                           int_env_pr
```

```
es) *
                                        pres_capac_per_v
ol + int_env_pres *
                                        air_leak_coefice
nt *
                                        ((1 - latenet_st
ructure_eclss) + 0) /
                                        efficiency_of_P
M)
            en_used_pres_new = max(0.0, min(available_e
n,
                                                en_needed
_pres))
            return en_needed_pres_new, \
                   en_used_pres_new

    return eclss_pressure

```

```
es) *
                                        pres_capac_per_v
ol + int_env_pres *
                                        air_leak_coefice
nt *
                                        ((1 - latenet_st
ructure_eclss) + 0) /
                                        efficiency_of_P
M)
            en_used_pres_new = max(0.0, min(available_e
n,
                                                en_needed
_pres))
            return en_needed_pres_new, \
                   en_used_pres_new

    return eclss_pressure

```