Formalization of Requirements for CPS and NTSS

Anonymous Author

1 Requirement Formalization

1.1 Autopilot

R1.4.1:

```
Textual:
Steady state roll commands shall be tracked within 1 degree.

Formalisation:
(forall tc in [ 1 , T ]:
(
  (  Tk(tc) != Tk(tc-1)  ) implies
  ( exists t2 in [ tc , T ]:
(
forall t in [ t2 , T ]:
(
 Phi(t) - PhiRef(t) <= 1
)
))));
```

R1.6:

```
Textual:
The maximum roll angle (Phi) allowed shall be 30 deg +/-10%

Formalisation:
forall t in [ 0 , T ]:
(
 Phi(t) <= 33  and  Phi(t) >= -33
)
);
```

R12.1:

```
Textual:
When the autopilot is enabled, the aircraft altitude should reach the desired altitude
within 500 seconds in calm air

Formalisation:
(forall tc in [ 1 , T ]:
(
(  APEng(tc) = 1  ) implies
( exists t2 in [ tc , T ]:
(
```

```
    forall t in [ t2 , T ]:
(
 alt(t) - ALTRef(t) >= 0
)
)
)
)
);
```

1.2 Finite State Machine

R1:

```
Textual:
Exceeding sensor  Limits shall latch an autopilot pullup when the pilot
is not in control (not Standby) and the system is Supported without failures (not Apfail).

Formalisation:
 (forall t in [ 0 , T ]:
(
(
 Standby(t) = 0  and  Apfail(t) = 0  and  Supported(t) = 1  and  Limits(t) = 1
) implies
 pullup(t) = 1
)
);
```

1.3 Non Linear Guidance

R6:

```
Textual:
The change in the magnitude of the output over one frame of execution with T sample period
shall not exceed the quantity of the combined velocity of the target plus the velocity
of the vehicle multiplied by T

Formalization:
(
    forall t in [ 1 , T ]:
(
(  xtarg[1](t) = xtarg[1](t-1)  and  xtarg[2](t) = xtarg[2](t-1)  and  xtarg[3](t) = xtarg[3](t-1)
and  yout[1](t) = Xap2[1](t)  and  yout[2](t) = Xap2[2](t)  and  yout[3](t) = Xap2[3](t)
and  yout[1](t-1) = Xap2[1](t-1)  and  yout[2](t-1) = Xap2[2](t-1)  and  yout[3](t-1) = Xap2[3](t-1)
)
implies

    (
            ( ( yout[1](t) - yout[1](t-1) ) * ( yout[1](t) - yout[1](t-1) ) ) +
            ( ( yout[2](t) - yout[2](t-1) ) * ( yout[2](t) - yout[2](t-1) ) ) +
            ( ( yout[3](t) - yout[3](t-1) ) * ( yout[3](t) - yout[3](t-1) ) )
```

```
    ) <= 0.0004 * (
            ( vv[1](t) + vt[1](t) ) * ( vv[1](t) + vt[1](t) ) +
            ( vv[2](t) + vt[2](t) ) * ( vv[2](t) + vt[2](t) ) +
            ( vv[3](t) + vt[3](t) ) * ( vv[3](t) + vt[3](t) )
    )

)
);
```

1.4 Regulator

R7:

```
Textual:
The Inner Loop Pitch Regulator Shall not command transient changes in angular roll
acceleration greater than 50 deg/sec2/sec

Formalisation:
(
    forall t in [ 1 , T ]:
(
 mcvdt_cmd_fcs_dps2(t) - mcvdt_cmd_fcs_dps2(t-1) <= 0.5
)
);
```

1.5 Tustin

R1:

```
Textual:
When Reset is True and the Initial Condition (ic) is bounded by the provided Top and
Bottom Limits (BL <= ic <= TL),
the Output (yout) shall equal the Initial Condition (ic).
Formalisation:

(
    forall t in [ 0 , T ]:
(
(  Reset(t) = 1  and  Ic(t) <= tl(t)  and  Ic(t) >= bl(t)  implies  yout(t) = Ic(t)  ) and
(  Reset(t) = 1  and  Ic(t) <= tl(t)  and  tl(t) >= bl(t)  implies  yout(t) = tl(t)  ) and
(  Reset(t) = 1  and  Ic(t) <= bl(t)  and  tl(t) >= bl(t)  implies  yout(t) = bl(t)  ) and
(  Reset(t) = 1  and  Ic(t) <= bl(t)  and  tl(t) < bl(t)  implies  yout(t) = bl(t)  )
)
);
```

R2:

```
Textual:
```

```
The Output (yout) shall be bounded by the provided Top and Bottom limits (TL and BL)
Formalisation:

(
    forall t in [ 0 , T ]:
(
(  tl(t) >= bl(t)  implies (  bl(t) <= yout(t)  and  yout(t) <= tl(t)  ) ) and
(  tl(t) < bl(t)  implies (  tl(t) <= yout(t)  and  yout(t) <= bl(t)  ) )
)
);
```

R4a:

```
Textual:
Over a 10 second computational duration at an execution frequency of 10 hz, the Output
should equal the sine of time t, sin, where time is defined as a vector from 0 to 10
by increments of 0.1 seconds within a +/- 0.1 tolerance for an input equal to the cosine
of time t, cos, with the sample delta time T = 0.1 seconds when in normal mode of operation.
Formalisation:

(
        forall t in [ 0 , T ]:
    (

    (
    ( tl(t) >= bl(t)  and  ( yout(t) >= bl(t)   and  yout(t) <= tl(t) )) or
( bl(t) >= tl(t)  and ( yout(t) >= tl(t)   and  yout(t) <= bl(t) ))
) implies
 |yout(T) - 10 | <= 0.1

)
);
```

R4b:

```
Textual:
Over a 10 second computational duration at an execution frequency of 10 hz, the Output
should equal the sine of time t, sin,where time is defined as a vector from 0 to 10 by
increments of 0.1 seconds within a +/- 0.1 tolerance for an input equal to the cosine of
time t, cos, with the sample delta time T = 0.1 seconds when in normal mode of operation.
Formalisation:

    (
        forall t in [ 0 , T ]:
    (

    (
    ( tl(t) >= bl(t)  and   yout(t) >= bl(t)   and  yout(t) <= tl(t) ) or
( bl(t) >= tl(t)  and  yout(t) >= tl(t)   and  yout(t) <= bl(t) )
) implies
 |yout(t) - sinsig(t) | <= 0.1
```

```
)
);
```

1.6 NTSS

Textual: Good network connectivity should be maintained even when high traffic flows through different priority classes.

Formalization:

$$\mathcal{R} = \begin{cases} \overline{mos}_n & \text{if } \bigwedge_{i=1..n} \overline{mos}_i < \overline{mosTh}_i, \\ 1 + \overline{mos}_{n-1} & \text{if } \overline{mos}_n \geq \overline{mosTh}_n \wedge \bigwedge_{i=1..n-1} \overline{mos}_i < \overline{mosTh}_i, \\ 2 + \overline{mos}_{n-2} & \text{if } \bigwedge_{i\in\{n-1,n\}} \overline{mos}_i \geq \overline{mosTh}_i \wedge \bigwedge_{i\in\{1..n-2\}} \overline{mos}_i < \overline{mosTh}_i, \\ \dots & \dots \\ n & \text{if } \bigwedge_{i=1..n} \overline{mos}_i \geq \overline{mosTh}_i \end{cases}$$

2 Input Specification for CPS and NTSS

Table 1: Input Specification for CPS and NTSS

Subject: TU1...TU9		
Input Name	Type	Range
`Xin`	Double	[-20,20]
`TL`	Double	[-10,10]
`BL`	Double	[-10,10]
`IC`	Double	[-20,20]

Subject: NLG		
Input Name	Type	Range
X_{targ}	Double	[-100,100]
X_v	Double	[-100,100]
V_v	Double	[-100,100]
V_{targ}	Double	[-100,100]
r	Double	[0,100]

Subject: REG		
Input Name	Type	Range
`beta_adc_deg`	Double	[0,5]
`vtas_adc_kts`	Double	[0,5]
`lev_md_fos_dps`	Double	[0,5]
`hdg_des_deg`	Double	[0,5]
`mev_emd_fes_dps`	Double	[0,5]
`alt_des_ft`	Double	[0,5]
`nev_cmd_fcs_dps`	Double	[0,5]
`xev_cmd_fcs_fps`	Double	[0,5]
`airspeed_des_fps`	Double	[0,5]
`hcv_cmd_fcs_fps`	Double	[0,5]
`lcv_fcs_dps`	Double	[0,5]
`mcv_fcs_dps`	Double	[0,5]
`ncv_fcs_dps`	Double	[0,5]
`dcv_fcs_fps`	Double	[0,5]
`zcv_cmd_fcs_fps`	Double	[0,5]
`betadot`	Double	[0,5]

Subject: FSM		
Input Name	Type	Range
`standby`	Boolean	{0,1}
`apfail`	Boolean	{0,1}
`supported`	Boolean	{0,1}
`limits`	Boolean	{0,1}

Subject: AP1...AP3		
Input Name	Type	Range
AP Eng	Boolean	{0,1}
HDG Mode	Boolean	{0,1}
ALT Mode	Boolean	{0,1}
HDG Ref	Double	[-180,180]
Turn knob	Double	[0,45]
ALT Ref	Double	[0,1000]
Pitch wheel	Double	[-30,30]
Throttle	Double	[0,1]

Subject: NTSS		
Input Name	Type	Range
`Class0`	Integer	[0,thresh_0]
`Class1`	Integer	[0,thresh_1]
`Class2`	Integer	[0,thresh_2]
`Class3`	Integer	[0,thresh_3]
`Class4`	Integer	[0,thresh_4]
`Class5`	Integer	[0,thresh_5]
`Class6`	Integer	[0,thresh_6]
`Class7`	Integer	[0,thresh_7]