DESIGN VERIFICATION OF CNS ALGORITHM IN SPARK ADA

DOCUMENT: [Filename]	DV_AdaModule_CNS_RevD_131115.doc
ORIGINATOR:	**Dr. Bob Davidov, Dr. Alex Deas**
DEPARTMENT:	**Verification Laboratory**
DATE UPDATED:	15th November 2013
REVISION:	**E**

APPROVALS	
___STG ___ Project Leader	15th November 2013 Date
___BK___ Quality Officer	15th November 2013 Date

Controlled Document ☒ Classified Document ☒ **DO NOT COPY.**

This is a controlled document under source control.

Quality High Integrity Environment

Certified at SIL 3

Revision History

Revision	Date	Description
A	10th Oct 2005	CNS Model, Review.
B	14th Jan 2006	Cleared for issue
C	2nd June 2008	(C1, 14th April 2006). 2^13 is defined instead of 10000 (C2, 18th April 2006). [15 15 19 24 30 38 46 57 65 70 76 83 91 101 114 164 303] /s is defined instead of [19 19 23 29 37 46 56 69 79 85 93 101 111 123 139 201 370] %/s The above update simplifies the actual calculation of the total CNS. Instead of dividing the SUM of dCNS by 10000 it is possible to use right shift. (C3, 2nd June 2008). Further accident data considered. Approved for customer release.
D	29th May 2013	Added CNS ADA code test coverage and test harness results for IEC 61508 approval review. D1 removed SPARK attributes from SPARK Ada code, to enable document to be published.
E	15th Nov 2013	Release includes the code version with handling of PPO2 > 1.6 ATM as a cubic ratio.

Table of Contents

1 Purpose and Scope

Central Nervous System Oxygen Toxicity (CNS) is one of two main forms of oxygen toxicity that affects divers: the other being pulmonary toxicity tracked as Oxygen Toxicity Units (OTUs).

OTU tracking forms a separate code module and is outside the scope of this document.

This document reports the formal modelling of CNS calculations followed by an implementation and verification of correctness of the Deep Life Ada module used in rebreathers and dive computers to determine CNS.

In that context, the scope of this document is broadened from a strict consideration of the formal model and its application to Deep Life's CNS code module, to include a brief discussion of appropriate safe CNS limits.

2 Summary

Central Nervous System oxygen toxicity (herein, CNS) is a well known phenomenon, whereby exposure to high levels of oxygen cause an acute oxygen poisoning. The human body can withstand only moderate levels of oxygen exposure.

Excessive oxygen exposure causes CNS symptoms of sudden narrowing of the visual acuity, sudden loss of consciousness with extremely violent spasms. The spasms are so violent that hard-hat divers have been known to throw off a helmet without undoing any latches, a feat that appears otherwise impossible from an examination of the helmet neck and the diver's head diameter. CNS spasms generally result in mortality when they occur to a diver underwater.

The CNS risk to divers is mitigated by measuring the exposure and indicating the exposure level to the diver: the diver is obliged to measures to keep the CNS exposure to within safe limits. The CNS risk is further mitigated by providing warnings and alarms in the event of a high PPO2.

Consideration is given to the scaling factors, or conservatism factors, that should be applied to enable a diver to keep the risk of oxygen convulsions at a "very unlikely" level. This is based on recent research on the antagonistic effects of depth, helium and carbon dioxide, as well as an analysis of the statistical risk based on the results of published trials using US Navy CNS tables and incident reports in sport diving.

NOAA proposed CNS exposure limits and a method for monitoring the exposure. Erik C. Baker [1] describes precise methods to calculate CNS to those NOAA limits as a function of the actual oxygen partial pressure (PPO_2) and the actual time of exposure. A model for the Baker equation is provided along with a simplified version which avoids the calculation of CNS fractions and logarithmic functions. The model is extended to cover PPO2 exposures above 1.6 ATA: the upper limit of PPO2 considered by NOAA.

3 SIL Assessment

The application has been assessed as SIL 3 for the computation and presentation, and SIL 2 for the hardware of dive computers, SIL 3 for the hardware of rebreathers where high PPO2 is a far more significant risk than on open circuit.

4 Requirement

The limits for oxygen exposure in the range zero to 1.6 ATM are taken from a study published in NOAA Repetitive Excursions (REPEX) Procedures Report [1].

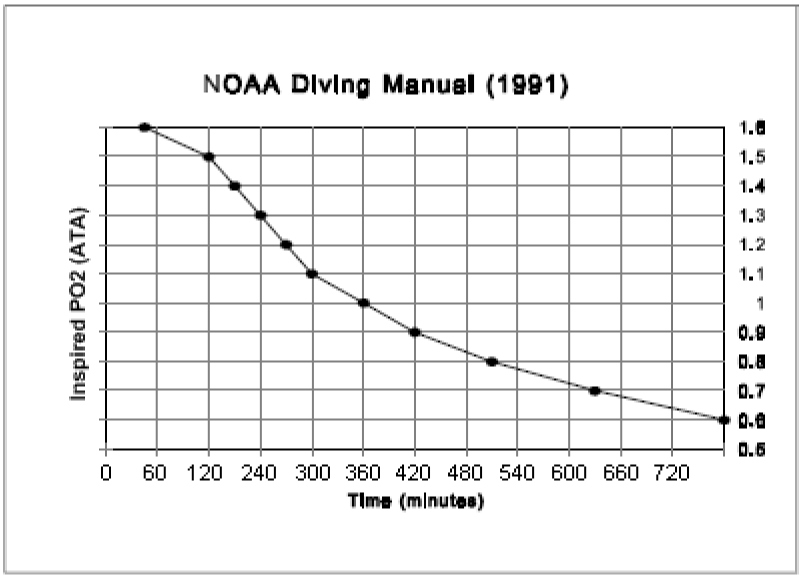


Figure 1. CNS limits established in the NOAA Repetitive Excursions (REPEX) Procedures Report [1].

For a constant depth profile (in which the PPO2 remains constant), the CNS fraction is calculated by

$$CNS\_fraction = \frac{time\_at\_PO2}{time\_Limit\_for\_PO2}$$

For an ascent or descent profile at a constant rate (where the PPO2 varies at a constant rate), the CNS fraction for a given PPO2 range with a linear time limit function can be calculated by

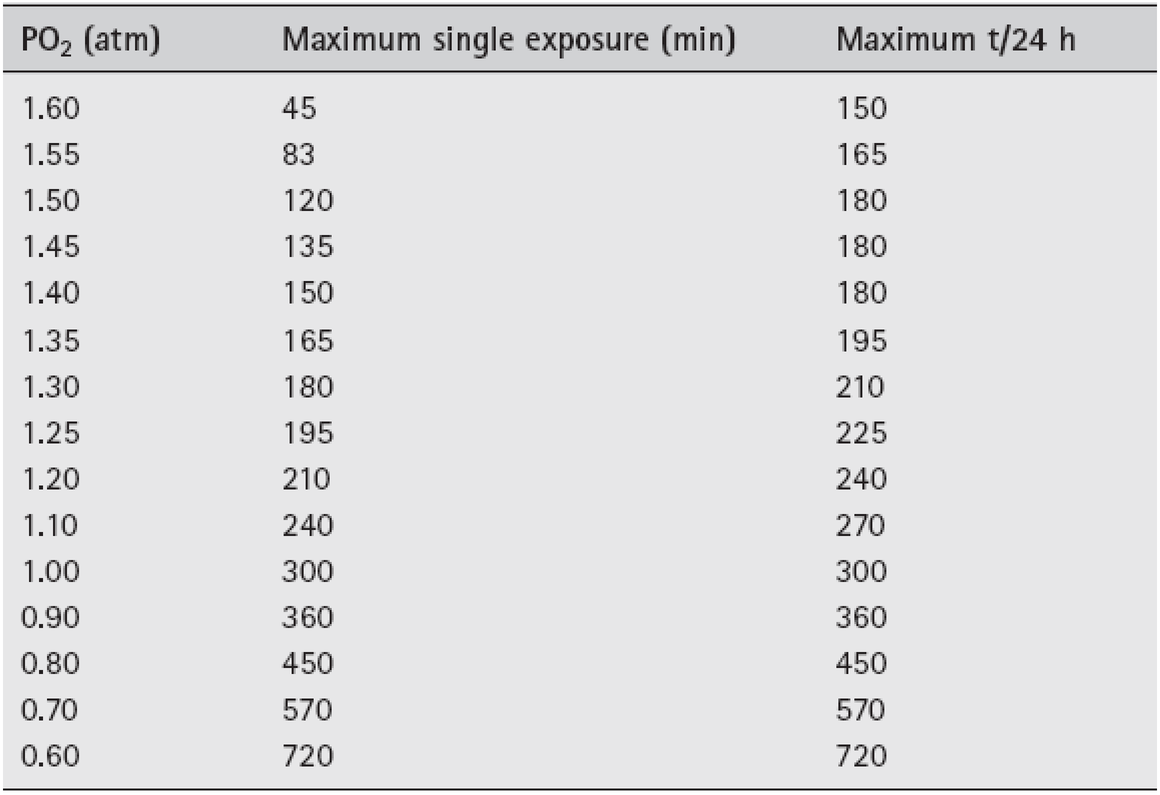
$$CNS\_fraction = \frac{\ln\left|T_{\lim} + mkt_x\right| * \ln\left|T_{\lim}\right|}{mk}$$

where m is the slope of the linear time limit function, k is the constant rate of change in PPO2 with

time, $T_{\lim}$ is the initial time limit for the interval shown in **Figure 1**, and t_x is the time of exposure over the interval.

These NOAA limits are tabulated overleaf.

Table 1 NOAA oxygen exposure limits. The Table gives limits in min fora single PPO2 exposure level, and for each day (24 h). (NOAA Diving Manual 2001, Figs 15.2, 16.2) [2]



PO_2 (atm)	Maximum single exposure (min)	Maximum t/24 h
1.60	45	150
1.55	83	165
1.50	120	180
1.45	135	180
1.40	150	180
1.35	165	195
1.30	180	210
1.25	195	225
1.20	210	240
1.10	240	270
1.00	300	300
0.90	360	360
0.80	450	450
0.70	570	570
0.60	720	720

4.1.1 CNS for PPO2 > 1.6 ATA

The reference papers do not indicate how PPO2 levels above 1.6 ATA should be handled.

There are four ways to handle CNS when the PPO2 is more than 1.6 ATA:

1. Flag an error and provide no calculation. This is not acceptable if a dive computer is used, as the diver then loses track of all CNS exposure. It is necessary to flag the error, but it is also necessary to perform a reasonable computation of CNS.
2. Extrapolate the NOAA curves beyond 1.6 ATM. In this case the diver rapidly achieves 100% CNS exposure, to the point where at even 1.65 ATM it is almost instant. This is not a useful function, as a diver may often peak the PPO2 and then it returns to normal and no untoward symptoms result.
3. Calculated the same CNS as for PPO2 of 1.6 ATA: this is rejected as it is unsafe – the diver certainly does not have a CNS clock of 45 minutes at a PPO2 of 4.0 ATA for example.
4. Reduce the CNS clock dramatically as a function of the PPO2 exposure.

   A function that is probably worst-case is to reduce the CNS clocks as a square of the differential, i.e.:

   Time for 100% CNS := 45/(((PPO2+0.1)-1.6)*10)^2

   An alternative function that is probably closer to the average diver's physiology uses a cubic ratio:

   Time for 100% CNS : 45 / PPO2^3 / 1.6^3

These functions give the exposures for a 100% CNS clock tabulated below and plotted overleaf.

Table 2. CNS Exposures for PPO2 > 1.6 using square and cubic ratio functions

PPO2 (ATM)	Square Law		Cubic Ratio	
	Acceleration Factor relative to PPO2 of 1.6 ATM	Exposure for 100% CNS, minutes	Acceleration Factor relative to PPO2 of 1.6 ATM	Exposure for 100% CNS, minutes
1.6	1	45	1.00	45.000
1.7	4	11.250	1.20	37.517
1.8	9	5.000	1.42	31.605
1.9	16	2.813	1.67	26.873
2	25	1.800	1.95	23.040
2.2	49	0.918	2.60	17.310
2.4	81	0.556	3.38	13.333
2.6	121	0.372	4.29	10.487
2.8	169	0.266	5.36	8.397
3	225	0.200	6.59	6.827
3.2	289	0.156	8.00	5.625
3.4	361	0.125	9.60	4.690
3.6	441	0.102	11.39	3.951
3.8	529	0.085	13.40	3.359
4	625	0.072	15.63	2.880
5	1225	0.037	30.52	1.475
10	7225	0.006	244.14	0.184

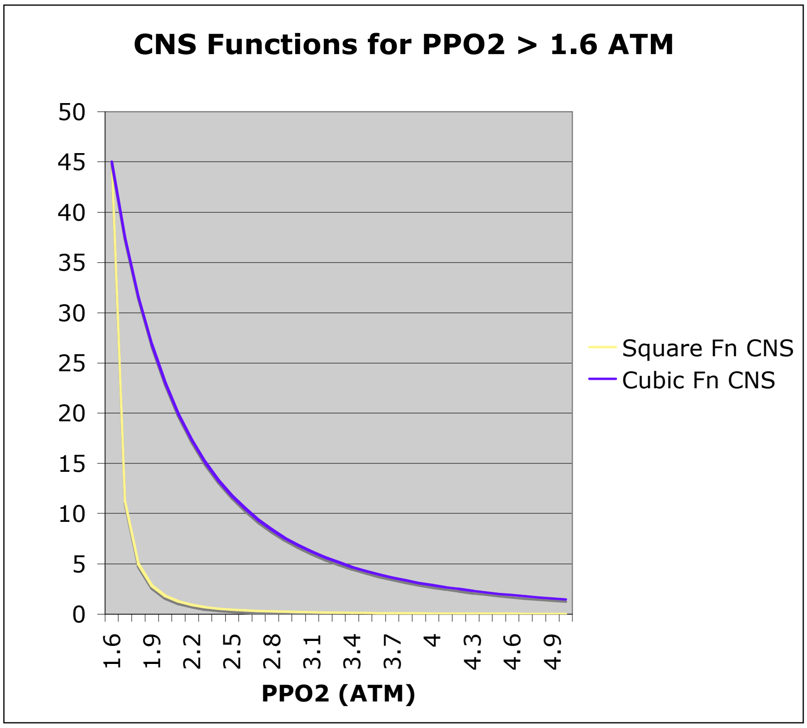


Figure 2. Time limit functions for PPO2 > 1.6 ATM

There is a paucity of scientific data to justify one factor over another: judgement should be applied and the decision that is made should be stated clearly to the user. The CNS clock for any individual diver almost certainly lays somewhere between these two curves.

Factors involving Carbon Dioxide exposure and aspects of rebreather use that may make an higher level of CNS attrition appropriate under some circumstances are discussed in the following sections, along with a means to apply that attrition.

4.1.2 Warnings and Alarms

The use of a high PPO2 should be apparent to the rebreather diver, as EN 14143:2003 (and 2013) contains the following requirements for warnings alarms in PPO2 monitors:

- Alarm if PPO2 > 2.0 bar[1] at any time.
- Alarm if PPO2 >= 1.6 bar for >= 1 minute
- Warning if PPO2 >= 1.6 bar

There is no equivalent requirement for dive computers or Open Circuit equipment: there is the anomaly that the European Community standard for dive computers, EN 13319, is not a PPE Standard, but dive computers clearly do perform a protective function so should be considered as PPE within the PPE Directive. At present, dive computers are generally CE certified without regard to PPE.

This present study is part of a Functional Safety case to IEC EN 61508 for a dive computer. IEC EN 61508 imposes far more stringent safety requirements on the equipment lifecycle than the PPE Directive. In that context, it is a requirement that where there is any risk of PPO2 exceeding safe levels, then the alarms and warnings apply. This means that addition to the high PPO2 warnings and alarms a high CNS warning shall be generated when the CNS exceeds 100%. The warnings and alarms shall be signalled regardless of whether the PPO2 levels are from a rebreather or from Open Circuit gases breathed under pressure.

5 THE ROLE OF CARBON DIOXIDE ON CNS

Oxygen toxicity limits must be reduced very considerably when the subject may have a high retained blood carbon dioxide level.

There have also been multiple diving accidents where a diver has convulsed, apparently from CNS, when the PPO2 has been far below the level normally needed for an oxygen convulsion but the diver has also been suffering from hypercapnia: the lowest on a rebreather was a PPO2 of just 1.0 ATM.

The risk factors that increase the retained CO2 on a rebreather and the known link between CO2 and reduced oxygen tolerance, means that there should be an extra degree of conservatism applied when oxygen is used in a rebreather as compared to used on open circuit as a Nitrox dive.

This is always a risk on a rebreather and can be a factor even on Open Circuit. There are at least nine different mechanisms that increase retained CO2 including:

- ♦ Scrubber breakthrough
- ♦ Scrubber bypass
- ♦ Scrubber tunnelling
- ♦ Scrubber CO2 regeneration

[1] The calibration of PPO2 in a rebreather is expressed in Bar in the standard, rather than ATM: for the purposes here Bar and ATM are considered equivalent (1 ATM is in fact 1.013 bar).

- Flapper valve failure
- Higher than normal Work of Breathing
- Exertion
- Divers who retain CO2 [23]
- Skip breathing.

A further reason for caution in applying oxygen toxicity limits is the large daily variation in sensitivity between individuals, and in any one individual over multiple days. The use of some common drugs is known to reduce the oxygen tolerance [10][11][12][13][14][15][16][19]. Rebreather divers are believed to be 5 years older on average than Open Circuit divers, and there is a greater probability that the diver is taking a medication that affects the oxygen toxicity for that individual.

6 NECESSARY DEGREE OF CONSERVATIVE MARGIN WHEN USED WITH REBREATHERS

The next issue for a formal model, is what down-rating is required for sports diving use. The Baker tables are based on US Navy tests, where just under 1% of divers have O2 toxicity symptoms when diving to 75% of the maximum CNS rating.

Appropriate limits for a diver are very different from limits in a dry recompression chamber. The occupants of a chamber will have a very low workload, will be warm and comfortable. The exposure may the thousands of times higher than for a sports diver in the water: some of the most extreme sports technical divers are even decompressing in a bell on such a high multiple.

Multiple factors indicate the Baker CNS limits are too high for general sports diving use, including:

1. Accidents. Six accidents involving O2 toxicity are reported on the RBW forum and there are indicators that the real number of incidents is much higher. Given the small number of rebreathers in use, this is level of risk is nowhere close to the requirements mandated for safety critical systems. The O2 convulsion incident level is estimated at between 1 in 10,000 hours and 1 in 100,000 hours in the CCR community, based on these reports alone. A PPO2 set point of 1.3 is the most commonly used: surveys on RBW refer.
2. Increased risk from deep diving [7], possibly due increased CO2 retention.
3. Increased risk from use of helium [6], due to mechanisms that are not understood.
4. Very highly increased risk from CO2 retention, either from higher CO2 inhalation as a result of scrubber breakthrough, scrubber bypass, mouthpiece volume, non-ideal flapper valves, or other forms of CO2 retention from Work of Breathing or CO2 retainers [5]. It is noted that sports divers are having CNS hits using PPO2s as low as 1.0, with just 25% of the standard Baker exposure: this is almost certainly due to these effects.
5. US Navy tests [8], [9]
6. The margin of accuracy of the PPO2 measurement.

Given these risk factors, it is recommended that the CNS algorithm be executed with 0.2 added to the measured PPO2. This means a PPO2 of 1.2 is treated like a PPO2 of 1.4 for the purposes of CNS calculation for the sports diver. For increased conservatism, figures of up to 0.3 could be added. This does extend dive times, particularly decompression time.

These figures are based on the small numbers of observed CNS accidents and diver-years exposure. The small numbers in these statistics (there is less than 10,000 diver-years of exposure, where a diver-year is the amount of sports diving a sports diver performs in a year).

It is recommended that the user have an option of aggressive CNS or conservative CNS: the former adding 0.2, the latter adding 0.3 to the measured PPO2.

The reduction in the probability of O2 toxicity from a given change in the PPO2 can be estimated based on the statistical curves published by Harabin et. Al [8] It is likely that adding 0.3 to the CNS calculation would reduce the incidence of O2 convulsions by 10^7, moving the incidence rate from to over 1 billion hours.

For commercial divers, where a full face helmet is used, the aggressive and conservative margins can be reduced, to 0.1 and 0.2 respectively. The latter is recommended.

The effect of the conservatism figures here, is to accelerate the CNS clock such that the chance of a sample appearing outside the distribution is probably less than 1 in 10^-9: again the limited numbers on which these statistics are based has to be recognised. This risk figure is based on extrapolations from available data and the data in the papers we have referenced. It does not prevent the diver using a higher PPO2 for short periods during decompression: the clock is an integral intended to indicate cumulative damage.

Consideration should be given to limiting the maximum PPO2 that can be set for dives below 20m to 1.1m, for the same reasons. The PPO2 may increase during decompression to 1.3, but there is very little gain above that: of course, the PPO2 is limited for the longest decompression stages (at 6msw to 3msw), by basic gas laws.

This simple means to provide CNS guard-banding for safety purposes, is trivial to implement. As such it has considerable merits over other methods that would change the whole form of the CNS limit curves.

7 Eric C. Baker Computation Method

The method in universal use by divers to “track” CNS oxygen toxicity is to compute a “CNS fraction” for each segment of the dive profile and then sum up these results to produce a total CNS fraction for the dive. These fractions are often multiplied by 100 and expressed as a percentage (CNS %). A CNS fraction is calculated by taking the time spent at a given PPO2 and dividing by the NOAA time limit for that PPO2. When the CNS fractions from all segments of the dive profile add up to 1 (or 100%) then the overall limit for CNS oxygen toxicity has been reached for that dive.

The NOAA curve of **Figure 1** calculated as the time limit function of PPO2 is shown in **Figure 3** where the PPO2 is [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6]; CNS time limit is [900 720 570 450 360 300 240 210 180 150 120 45].

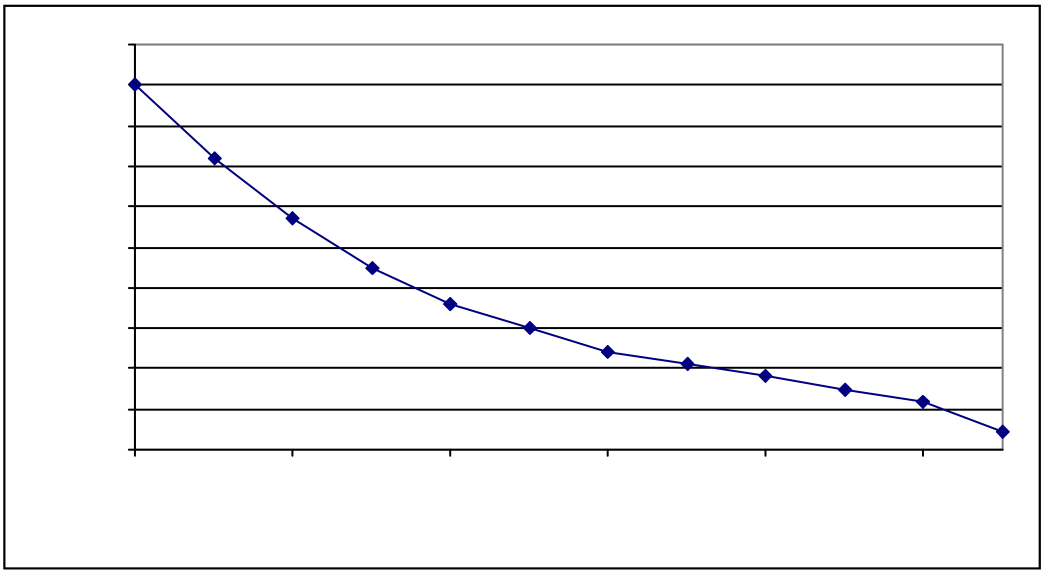


Figure 3. Time limit function of PPO2 calculated using Eric Baker method

Erik C. Baker made one extrapolation to describe a time limit equation for the PPO2 range from 0.5 to 0.6 in order to complete the series. An individual equation is written as:

$$Time\_Limit = m * PPO2 + b,$$

where m is the slope of the line and b is the intercept for the given PPO2 range (**Table 3**).

Table 3. Coefficients for the series of linear equations

PPO2, ATA	m (slope)	b (intercept)
0.5 - 0.6	-1800	1800
0.6 - 0.7	-1500	1620
0.7 - 0.8	-1200	1410
0.8 - 0.9	-900	1170
0.9 - 1.1	-600	900
1.1 - 1.5	-300	570
1.5 - 1.6	-750	1245

Calculation of $Time\_Limit$ within a PPO2 range applies when the present PPO2 is greater than the lower limit of the PPO2 range and is less than or equal to the upper limit of the PPO2 range.

Note: The above equations of Eric C. Baker method combine the data points of the NOAA Diving Manual (1991) (see **Figure 1**) and NOAA Diving Manual (2001) [**Table 1**] for CNS oxygen toxicity and provide the identical results for the data of both manuals.

7.1 FORTRAN Base Code

The CNS algorithm and its FORTRAN realisation is described by Erik C. Baker, P.E. in [1].

The calculations are based on diving at sea level:

1. Determine segment time for the ascent or descent profile (at a constant rate):

   Segment Time = (Final Depth - Starting Depth) ¸ Rate* of ascent or descent

   * Note: descent rate should be expressed as positive number (i.e. +60 fsw/min) while ascent rate should be expressed as a negative number (i.e. -30 fsw/min).

2. Determine initial and final pressures of the profile in atmospheres absolute (ATA):

   Initial Pressure (ATA) = (Starting Depth + 33) ¸ 33
   Final Pressure (ATA) = (Final Depth + 33) ¸ 33

3. Of the initial and final pressures (ATA), determine which is maximum and which is minimum:

   Maximum Pressure (ATA) = Max(Initial Pressure, Final Pressure)
   Minimum Pressure (ATA) = Min(Initital Pressure, Final Pressure)

4. Determine the maximum PPO2 and minimum PPO2 of the profile:

   Maximum PPO2 = Maximum Pressure (ATA) ´ Fraction of O2 in breathing gas
   Minimum PPO2 = Minimum Pressure (ATA) ´ Fraction of O2 in breathing gas

5. Determine if the maximum and minimum PPO2's fall within the range for oxygen toxicity (i.e. greater than 0.5) and assign a new variable (Low PPO2) to describe the lowest PPO2 of the profile that is within the range for oxygen toxicity:

   If Maximum PPO2 < 0.5 then [calculation does not apply]
   If Minimum PPO2 < 0.5 then Low PPO2 = 0.5, else Low PPO2 = Minimum PPO2

6. Determine time of exposure within range for oxygen toxicity:

$$\text{Exposure Time} = \text{Segment Time}\,\frac{\text{Maximum PO}_2 * \text{Low PO}_2}{\text{Maximum PO}_2 * \text{Minimum PO}_2}$$

In the FORTRAN programming language, the above steps can be written as:

```
SGTIME = (FDEPTH - SDEPTH)/RATE
IPATA = (SDEPTH + 33.0)/33.0
FPATA = (FDEPTH + 33.0)/33.0
MAXATA = MAX(IPATA, FPATA)
MINATA = MIN(IPATA, FPATA)
MAXPO2 = MAXATA * FO2(GASMIX)
MINPO2 = MINATA * FO2(GASMIX)
IF (MAXPO2 .LE. 0.5) GOTO 000 [exit this calculation sequence]
IF (MINPO2 .LT. 0.5) THEN
LOWPO2 = 0.5
ELSE
LOWPO2 = MINPO2
END IF
O2TIME = SGTIME*(MAXPO2 - LOWPO2)/(MAXPO2 - MINPO2)

DO 10 I = 1,7
IF ((MAXPO2 .GT. PO2LO(I)) .AND. (LOWPO2 .LE. PO2HI(I))) THEN
IF ((MAXPO2 .GE. PO2HI(I)) .AND. (LOWPO2 .LT. PO2LO(I))) THEN
IF (SDEPTH .GT. FDEPTH) THEN
PO2O(I) = PO2HI(I)
PO2F(I) = PO2LO(I)
ELSE
PO2O(I) = PO2LO(I)
PO2F(I) = PO2HI(I)
END IF
SEGPO2(I) = PO2F(I) - PO2O(I)
ELSE IF ((MAXPO2 .LT. PO2HI(I)) .AND. (LOWPO2 .LE. PO2LO(I))) THEN
IF (SDEPTH .GT. FDEPTH) THEN
```

```
PO2O(I) = MAXPO2
PO2F(I) = PO2LO(I)
ELSE
PO2O(I) = PO2LO(I)
PO2F(I) = MAXPO2
END IF
SEGPO2(I) = PO2F(I) - PO2O(I)
ELSE IF ((LOWPO2 .GT. PO2LO(I)) .AND. (MAXPO2 .GE. PO2HI(I))) THEN
IF (SDEPTH .GT. FDEPTH) THEN
PO2O(I) = PO2HI(I)
PO2F(I) = LOWPO2
ELSE
PO2O(I) = LOWPO2
PO2F(I) = PO2HI(I)
END IF
SEGPO2(I) = PO2F(I) - PO2O(I)
ELSE
IF (SDEPTH .GT. FDEPTH) THEN
PO2O(I) = MAXPO2
PO2F(I) = LOWPO2
ELSE
PO2O(I) = LOWPO2
PO2F(I) = MAXPO2
END IF
SEGPO2(I) = PO2F(I) - PO2O(I)
END IF
OTIME(I) = O2TIME*ABS(SEGPO2(I))/(MAXPO2 - LOWPO2)
ELSE
OTIME(I) = 0.0
END IF
10 CONTINUE
DO 20 I = 1,7
IF (OTIME(I) .EQ. 0.0) THEN
CNS(I) = 0.0
GOTO 20
ELSE
TLIMI(I) = LIMSLP(I)*PO2O(I) + LIMINT(I)
MK(I) = LIMSLP(I)*(SEGPO2(I)/OTIME(I))
CNS(I) = 1.0/MK(I)*(LOG(ABS(TLIMI(I) + MK(I)*OTIME(I))) -
* LOG(ABS(TLIMI(I))))
END IF
20 CONTINUE
SUMCNS = 0.0
DO 30 I = 1,7
TMPCNS = SUMCNS
SUMCNS = TMPCNS + CNS(I)
30 CONTINUE
```

Example Calculation of Eric C. Baker [1]:

Consider the following dive profile using EAN 32 for the gas mix (ignore decompression considerations for this example):

Segment 1 Descent from 0 fsw to 120 fsw at 40 fsw/min

Segment 2 Constant depth at 120 fsw for 22 min

Segment 3 Long, slow ascent from 120 fsw to 0 fsw at -4 fsw/min

What is the CNS fraction accumulated on this dive? Program solution:

For Segment 1Program calculates a CNS fraction = .0090

For Segment 2Program calculates a CNS fraction = .1761

For Segment 3Program calculates a CNS fraction = .0895

Total CNS fraction for this dive = .0090 + .1761 + .0895 **= .2746 = 27.5%**

Note: If PPO2 < 0.5 then the calculation does not apply

7.2 MATLAB SIMULINK CNS MODEL

Let's consider, for example, the CNS for the constant PPO2 of 1.0 using the time limit function of PPO2 shown in **Figure 3**. The time limit for unity PPO2 is 300 min. It means that CNS reaches 10, 50 and 100% in 30, 150 and 300 minutes respectively. The weight of CNS increases at 1/3 % each minute. The second fraction of CNS is calculated as

$$\Delta CNS(PPO2) = \frac{100}{60 * Time\_Limit\_for\_PPO2}$$

The ΔCNS function of PPO2 in %/s is shown in **Figure 4** where the PPO2 is [0 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.25 1.3 1.35 1.4 1.45 1.5 1.55 1.6] ATA; ΔCNS is [1/540 1/540 1/432 1/342 1/270 1/216 1/180 1/144 1/126 1/117 1/108 1/99 1/90 1/81 1/72 10/498 1/27] %/s. CNS fractions of PPO2 < 0.5 are zero.

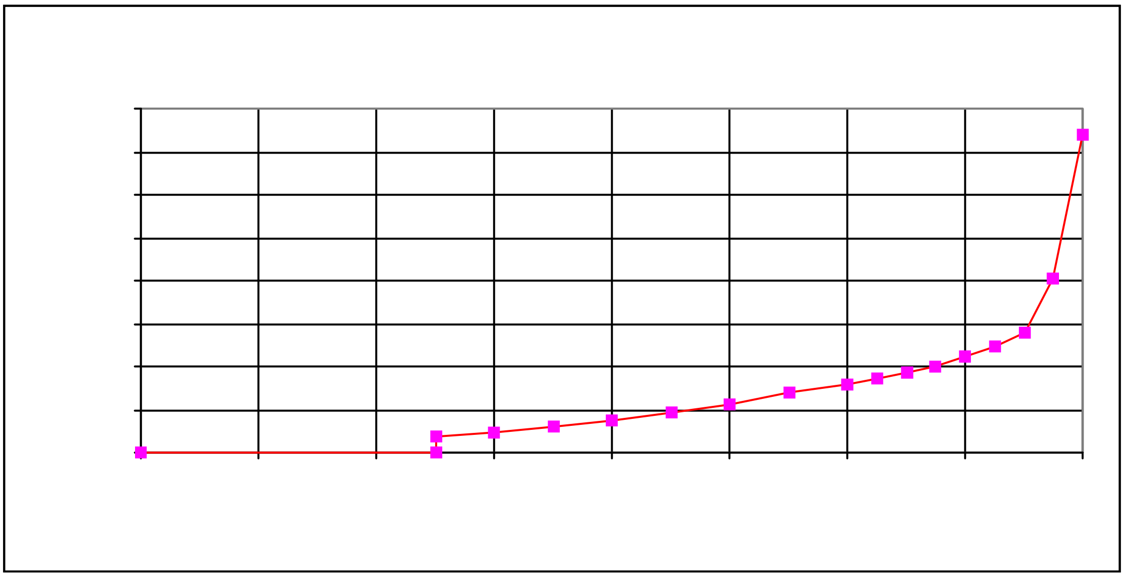


Figure 4. CNS fraction as function of PPO2 in %/s

The CNS as function of PPO2 and time can be calculated as

$$CNS(PPO2,t) = \sum \Delta CNS(PPO2(n * \Delta t)),$$

where $t = n * \Delta t$ is time of diving, Δt is 1s clock.

Calculation of CNS in the MatLAB simulink is shown in **Figure 5**. The output of the saturation block: "PPO2<1.6" is from 0 to 1.6. The discontinuity block: "dCNS" is ΔCNS function of PPO2. The output of the switch: "Threshold=0.5" is zero when the PPO2 is less than 0.5 and the switch output is equal to the output of the dCNS block when the PPO2 is more or equal to 0.5. The saturation block of "CNS<100%" limits the model output inside 100%.

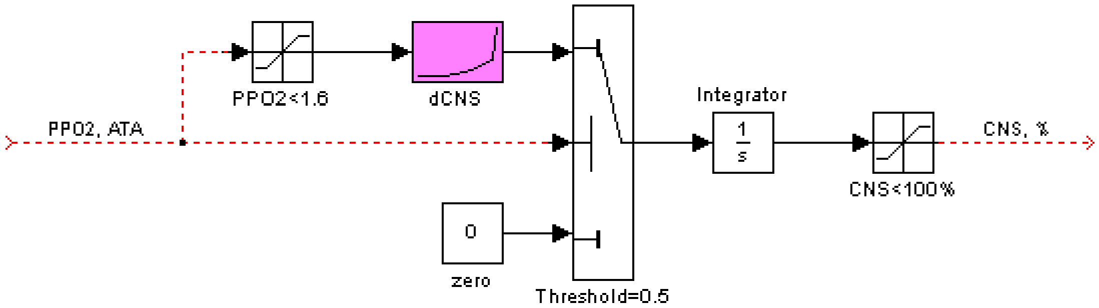


Figure 5. CNS model in MatLAB simulink

7.3 CNS CALCULATION USING INTEGER NUMBERS

In safety critical systems, integer implementations are generally used instead of floating point numbers, to decrease the calculation time, simplify the structures required, for example, for FPGA implementations, to enable formal proofs of correctness to be used and to enable adequate fault coverage with a reasonable sized test harness.

To preserve CNS accuracy the vector of ΔCNS would normally be multiplied 10000 times and then the sum of the CNS fractions must be divided in the same amount.

The integer ΔCNS function of PPO2 per second is:

PPO2 = [0 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.25 1.3 1.35 1.4 1.45 1.5 1.55 1.6] ATA;
ΔCNS =[15 15 19 24 30 38 46 57 65 70 76 83 91 101 114 164 303] /s.

7.4 MATLAB M- CODE

7.4.1 TRANSMISSION FROM FORTRAN INTO M-CODE FORMAT

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CNS_base.m     v1.0a
% MatLAB 2012a
% Bob Davidov, Alex Deas
% 14 May 2013
%
% NOTE: transmission of CNS from FORTRAN to MatLAB
% Source of FORTRAN code is Erik C. Baker, P.E. "Oxygen Toxicity
% Calculations", cns.pdf, pp. 7..12
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% input data
FO2 = [0.32];   % 0.32 for Segment 1, 2 and 3
GASMIX = 1;     % 0.32 for Segment 1, 2 and 3
FDEPTH = 120;   % [0 120 120] for Segment 1, 2 and 3
SDEPTH = 120;   % [0 120 120] for Segment 1, 2 and 3
RATE = -4;      % [40 -4] for Segment 1 and 3
TIME = 22;      % for Segment 2

PO2LO = [  0.5   0.6   0.7  0.8  0.9  1.1  1.5]; % low end of PPO2 range
PO2HI = [  0.6   0.7   0.8  0.9  1.1  1.5  1.6]; % high end of PPO2 range
LIMSLP  = [-1800 -1500 -1200 -900 -600 -300 -750]; % slope coefficient for
PPO2 range
LIMINT  = [1800   1620  1410 1170  900  570 1245]; % intercept coefficient
for PPO2 range

CNS(1:7) = 0;
```

```
if FDEPTH == SDEPTH
    FPATA = (FDEPTH + 33.0)/33.0;
    PPO2 = FPATA * FO2(GASMIX);
    if PPO2 > 0.5
        for I = 1:7
            if (PPO2 > PO2LO(I)) && (PPO2 <= PO2HI(I))
                TLIM = LIMSLP(I)*PPO2 + LIMINT(I);
                CNS = TIME/TLIM;
            end
        end
    else
        CNS = 0;
    end
else
    SGTIME = (FDEPTH - SDEPTH)/RATE;
    IPATA = (SDEPTH + 33.0)/33.0;
    FPATA = (FDEPTH + 33.0)/33.0;
    MAXATA = max(IPATA, FPATA);
    MINATA = min(IPATA, FPATA);
    MAXPO2 = MAXATA * FO2(GASMIX);
    MINPO2 = MINATA * FO2(GASMIX);
    if MAXPO2 > 0.5
        if MINPO2 < 0.5
            LOWPO2 = 0.5
        else
            LOWPO2 = MINPO2
        end
        O2TIME = SGTIME*(MAXPO2 - LOWPO2)/(MAXPO2 - MINPO2);

        for I = 1:7
            if ((MAXPO2 > PO2LO(I)) && (LOWPO2 <= PO2HI(I)))
                if ((MAXPO2 >= PO2HI(I)) && (LOWPO2 < PO2LO(I)))
                    if (SDEPTH > FDEPTH)
                        PO2O(I) = PO2HI(I);
                        PO2F(I) = PO2LO(I);
                    else
                        PO2O(I) = PO2LO(I);
                        PO2F(I) = PO2HI(I);
                    end
                    SEGPO2(I) = PO2F(I) - PO2O(I);
                elseif ((MAXPO2 < PO2HI(I)) && (LOWPO2 <= PO2LO(I)))
                    if (SDEPTH > FDEPTH)
                        PO2O(I) = MAXPO2;
                        PO2F(I) = PO2LO(I);
                    else
                        PO2O(I) = PO2LO(I);
                        PO2F(I) = MAXPO2;
                    end
                    SEGPO2(I) = PO2F(I) - PO2O(I);
                elseif ((LOWPO2 > PO2LO(I)) && (MAXPO2 >= PO2HI(I)))
                    if (SDEPTH > FDEPTH)
                        PO2O(I) = PO2HI(I);
                        PO2F(I) = LOWPO2;
                    else
                        PO2O(I) = LOWPO2;
                        PO2F(I) = PO2HI(I);
                    end
                    SEGPO2(I) = PO2F(I) - PO2O(I);
                else
                    if (SDEPTH > FDEPTH)
```

```
                        PO2O(I) = MAXPO2;
                        PO2F(I) = LOWPO2;
                    else
                        PO2O(I) = LOWPO2;
                        PO2F(I) = MAXPO2;
                    end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                end
                OTIME(I) = O2TIME * abs(SEGPO2(I))/(MAXPO2 - LOWPO2);
            else
                OTIME(I) = 0.0;
            end
        end
        for I = 1:7
            if OTIME(I) == 0.0
                CNS(I) = 0.0;
            else
                TLIMI(I) = LIMSLP(I)*PO2O(I) + LIMINT(I);
                MK(I) = LIMSLP(I)*(SEGPO2(I)/OTIME(I));
                CNS(I) = 1.0/MK(I)*(log(abs(TLIMI(I) + MK(I)*OTIME(I))) -
log(abs(TLIMI(I))))
            end
        end
        SUMCNS = 0.0;
        for I = 1:7
            TMPCNS = SUMCNS;
            SUMCNS = TMPCNS + CNS(I);
        end
    end
end

% End of m-file
```

7.4.2 MATLAB CODE FOR PREDICT MODE

```
clear all

profile_depth = [0 120 120 0] * 10 /33;
profile_time = [0 3 22+3 22+3+120/4];
CNS = 0;
for i = 2:length(profile_depth)
    Depth_Old_m = profile_depth (i);
    Depth_m = profile_depth (i-1);
    Gas_percent = 32;
    Sample_Time_min = profile_time (i) - profile_time (i-1);

    CNS_inc = CNS_function(Depth_Old_m, Depth_m, Gas_percent,
Sample_Time_min);
    CNS = CNS + CNS_inc;
end
CNS_percent = CNS * 100;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CNS_function.m     v1.0a
% MatLAB 2012a
% Bob Davidov, Alex Deas
% 14 May 2013
%
% NOTE: transmission of CNS from FORTRAN to MatLAB
% Source of FORTRAN code is Erik C. Baker, P.E. "Oxygen Toxicity
% Calculations", cns.pdf, pp. 7..12
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function CNS_inc = CNS_function(Depth_Old_m, Depth_m, Gas_percent,
Sample_Time_min)

FO2 = Gas_percent / 100;

CNS_inc = 0;
CNS(1:7) = 0;

PO2LO = [  0.5   0.6   0.7  0.8  0.9  1.1  1.5]; % low end of PPO2 range
PO2HI = [  0.6   0.7   0.8  0.9  1.1  1.5  1.6]; % high end of PPO2 range
LIMSLP  = [-1800 -1500 -1200 -900 -600 -300 -750]; % slope coefficient for
PPO2 range
LIMINT  = [1800   1620  1410 1170  900  570 1245]; % intercept coefficient
for PPO2 range

if Depth_m == Depth_Old_m
    FPATA = (Depth_m + 10.0)/10.0;
    PPO2 = FPATA * FO2;
    if PPO2 > 1.6
        PPO2 = 1.6;
    end
    if PPO2 > 0.5
        for I = 1:7
            if (PPO2 > PO2LO(I)) && (PPO2 <= PO2HI(I))
                TLIM = LIMSLP(I)*PPO2 + LIMINT(I);
                CNS_inc = Sample_Time_min/TLIM;
            end
        end
    else
        CNS_inc = 0;
    end
else
    IPATA = (Depth_Old_m + 10.0)/10.0;
    FPATA = (Depth_m + 10.0)/10.0;
    MAXATA = max(IPATA, FPATA);
    MINATA = min(IPATA, FPATA);
    MAXPO2 = MAXATA * FO2;
    if MAXPO2 > 1.6
        MAXPO2 = 1.6;
    end
    MINPO2 = MINATA * FO2;
    if MINPO2 > 1.6
        MINPO2 = 1.6;
    end
    if MAXPO2 == MINPO2
        TLIM = -750*1.6 + 1245;
        CNS_inc = Sample_Time_min/TLIM;
    else
        if MAXPO2 > 0.5
            if MINPO2 < 0.5
                LOWPO2 = 0.5;
            else
                LOWPO2 = MINPO2;
            end
            O2TIME = Sample_Time_min*(MAXPO2 - LOWPO2)/(MAXPO2 - MINPO2);

            for I = 1:7
                if ((MAXPO2 > PO2LO(I)) && (LOWPO2 <= PO2HI(I)))
```

```
                    if ((MAXPO2 >= PO2HI(I)) && (LOWPO2 < PO2LO(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = PO2HI(I);
                            PO2F(I) = PO2LO(I);
                        else
                            PO2O(I) = PO2LO(I);
                            PO2F(I) = PO2HI(I);
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    elseif ((MAXPO2 < PO2HI(I)) && (LOWPO2 <= PO2LO(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = MAXPO2;
                            PO2F(I) = PO2LO(I);
                        else
                            PO2O(I) = PO2LO(I);
                            PO2F(I) = MAXPO2;
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    elseif ((LOWPO2 > PO2LO(I)) && (MAXPO2 >= PO2HI(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = PO2HI(I);
                            PO2F(I) = LOWPO2;
                        else
                            PO2O(I) = LOWPO2;
                            PO2F(I) = PO2HI(I);
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    else
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = MAXPO2;
                            PO2F(I) = LOWPO2;
                        else
                            PO2O(I) = LOWPO2;
                            PO2F(I) = MAXPO2;
                        end
                            SEGPO2(I) = PO2F(I) - PO2O(I);
                    end
                    OTIME(I) = O2TIME * abs(SEGPO2(I))/(MAXPO2 - LOWPO2);
                else
                    OTIME(I) = 0.0;
                end
            end
            for I = 1:7
                if OTIME(I) == 0.0
                    CNS(I) = 0.0;
                else
                    TLIMI(I) = LIMSLP(I)*PO2O(I) + LIMINT(I);
                    MK(I) = LIMSLP(I)*(SEGPO2(I)/OTIME(I));
                    CNS(I) = 1.0/MK(I)*(log(abs(TLIMI(I) + MK(I)*OTIME(I)))
- log(abs(TLIMI(I))));
                end
            end
            CNS_inc = sum (CNS);
        end
    end
end

% End of m-file
```

7.4.3 MatLAB CNS code for RT application

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% main.m     v1.0a
% MatLAB 2012a
% Bob Davidov, Alex Deas
% 14 May 2013
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

%profile_depth = [0 120 120 0] * 10 /33; % in fsw
profile_depth = [0 6 6 0]; % in msw
profile_time = [0 3 100 103];
Gas_percent = 100;

CNS = 0;
Sample_Time_min = 1; %/60; % /60 in sec
Depth_Old_m = profile_depth (1);
Time = profile_time (1);
i = 2;
CNS(1) = 0;
Time_plot(1) = 0;
j = 2;
while Time < profile_time (length (profile_time))

    Time = Time + Sample_Time_min;
    if Time <= profile_time (i)
        Depth_m = Depth_Old_m + ((profile_depth(i)-profile_depth(i-
1))/(profile_time(i)-profile_time(i-1)))*Sample_Time_min;
        CNS_inc = CNS_function (Depth_Old_m, Depth_m, Gas_percent,
Sample_Time_min);
        CNS(j) = CNS(j-1) + CNS_inc;
        Time_plot(j) = Time;
        j = j + 1;
        Depth_Old_m = Depth_m;
    else
        i = i+1;
%         Depth_m = Depth_Old_m + ((profile_depth(i)-profile_depth(i-
1))/(profile_time(i)-profile_time(i-1)))*Sample_Time_min;
    end
end
CNS_percent = CNS * 100;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% CNS_function.m     v1.0a
% MatLAB 2012a
% Bob Davidov, Alex Deas
% 14 May 2013
%
% NOTE: transmission of CNS from FORTRAN to MatLAB
% Source of FORTRAN code is Erik C. Baker, P.E. "Oxygen Toxicity
% Calculations", cns.pdf, pp. 7..12
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
function CNS_inc = CNS_function(Depth_Old_m, Depth_m, Gas_percent,
Sample_Time_min)

FO2 = Gas_percent / 100;
```

```
CNS_inc = 0;
CNS(1:7) = 0;

PO2LO = [  0.5   0.6   0.7  0.8  0.9  1.1  1.5]; % low end of PPO2 range
PO2HI = [  0.6   0.7   0.8  0.9  1.1  1.5  1.6]; % high end of PPO2 range
LIMSLP  = [-1800 -1500 -1200 -900 -600 -300 -750]; % slope coefficient for
PPO2 range
LIMINT  = [1800   1620  1410 1170  900  570 1245]; % intercept coefficient
for PPO2 range

if Depth_m == Depth_Old_m
    FPATA = (Depth_m + 10.0)/10.0;
    PPO2 = FPATA * FO2;
    if PPO2 > 1.6
        PPO2 = 1.6;
    end
    if PPO2 > 0.5
        for I = 1:7
            if (PPO2 > PO2LO(I)) && (PPO2 <= PO2HI(I))
                TLIM = LIMSLP(I)*PPO2 + LIMINT(I);
                CNS_inc = Sample_Time_min/TLIM;
            end
        end
    else
        CNS_inc = 0;
    end
else
    IPATA = (Depth_Old_m + 10.0)/10.0;
    FPATA = (Depth_m + 10.0)/10.0;
    MAXATA = max(IPATA, FPATA);
    MINATA = min(IPATA, FPATA);
    MAXPO2 = MAXATA * FO2;
    if MAXPO2 > 1.6
        MAXPO2 = 1.6;
    end
    MINPO2 = MINATA * FO2;
    if MINPO2 > 1.6
        MINPO2 = 1.6;
    end
    if MAXPO2 == MINPO2
        TLIM = -750*1.6 + 1245;
        CNS_inc = Sample_Time_min/TLIM;
    else
        if MAXPO2 > 0.5
            if MINPO2 < 0.5
                LOWPO2 = 0.5;
            else
                LOWPO2 = MINPO2;
            end
            O2TIME = Sample_Time_min*(MAXPO2 - LOWPO2)/(MAXPO2 - MINPO2);

            for I = 1:7
                if ((MAXPO2 > PO2LO(I)) && (LOWPO2 <= PO2HI(I)))
                    if ((MAXPO2 >= PO2HI(I)) && (LOWPO2 < PO2LO(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = PO2HI(I);
                            PO2F(I) = PO2LO(I);
                        else
                            PO2O(I) = PO2LO(I);
                            PO2F(I) = PO2HI(I);
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    elseif ((MAXPO2 < PO2HI(I)) && (LOWPO2 <= PO2LO(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = MAXPO2;
                            PO2F(I) = PO2LO(I);
                        else
                            PO2O(I) = PO2LO(I);
                            PO2F(I) = MAXPO2;
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    elseif ((LOWPO2 > PO2LO(I)) && (MAXPO2 >= PO2HI(I)))
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = PO2HI(I);
                            PO2F(I) = LOWPO2;
                        else
                            PO2O(I) = LOWPO2;
                            PO2F(I) = PO2HI(I);
                        end
                        SEGPO2(I) = PO2F(I) - PO2O(I);
                    else
                        if (Depth_Old_m > Depth_m)
                            PO2O(I) = MAXPO2;
                            PO2F(I) = LOWPO2;
                        else
                            PO2O(I) = LOWPO2;
                            PO2F(I) = MAXPO2;
                        end
                            SEGPO2(I) = PO2F(I) - PO2O(I);
                    end
                    OTIME(I) = O2TIME * abs(SEGPO2(I))/(MAXPO2 - LOWPO2);
                else
                    OTIME(I) = 0.0;
                end
            end
            for I = 1:7
                if OTIME(I) == 0.0
                    CNS(I) = 0.0;
                else
                    TLIMI(I) = LIMSLP(I)*PO2O(I) + LIMINT(I);
                    MK(I) = LIMSLP(I)*(SEGPO2(I)/OTIME(I));
                    CNS(I) = 1.0/MK(I)*(log(abs(TLIMI(I) + MK(I)*OTIME(I)))
- log(abs(TLIMI(I))));
                end
            end
            CNS_inc = sum (CNS);
        end
    end
end

% End of m-file
```

8 VERIFICATION RESULTS

8.1 REFERENCE TEST RESULT

Reference CNS test results shown in [1] were checked.

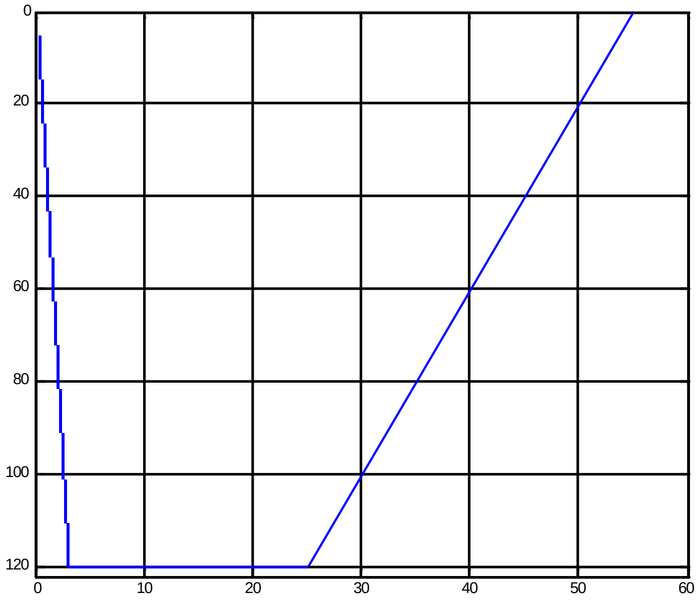


Figure 6. Reference dive profile.

8.2 MATLAB CNS TEST RESULT

MatLAB code shown in the item [7.4.1] provides the CNS as it is in the reference FORTRAN CNS response against the same input data:

Total CNS fraction for this dive = .0090 + .1761 + .0895 = .2746 = 27.5%

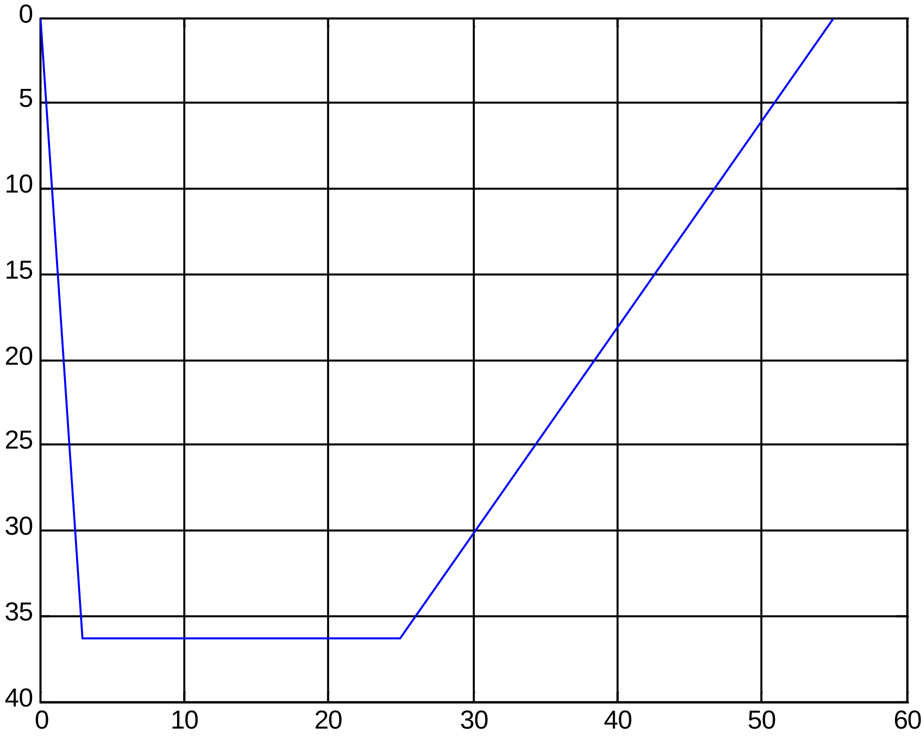


Figure 7. Applied dive profile

MatLAB CNS script operated in the predictive mode with

```
profile_depth = [0 120 120 0] * 10 /33;
profile_time = [0 3 22+3 22+3+120/4];
Depth_Old_m = profile_depth (i);
Depth_m = profile_depth (i-1);
Gas_percent = 32;
Sample_Time_min = profile_time (i) - profile_time (i-1);
```

calculated the Total CNS fraction as .0090 + .1761 + .0895 = .2746 = 27.5%. It is exactly as the test result of the reference FORTRAN script.

MatLAB CNS script operated in the predictive mode with

```
profile_depth = [0 120 120 0] * 10 /33;
profile_time = [0 3 22+3 22+3+120/4];
Depth_Old_m = profile_depth (i);
Depth_m = profile_depth (i-1);
Gas_percent = 32;
Sample_Time_min = profile_time (i) - profile_time (i-1);
```

calculated the Total CNS fraction as .0090 + .1761 + .0895 = .2746 = 27.5%. It is exactly as the test result of the reference FORTRAN script.

Result CNS of MatLAB script operated in RT mode does not corresponds to the CNS reference results. The differences are

CNS reference for 3 min dive from 0 to 120 fsw is 0.009 or 0.9%

When the same CNS reference script shows CNS of 0.97 or 0.97% for 3 steps from 0 to 40 msw, from 40 to 80 msw and from 80 to 120 msw for one min each step. The CNS error for 3 min ascent dive is 7%. The source of the error is the CNS function is calculated in the table polynomial mode but not in the analytical mode.

MatLAB CNS script in RT mode shows the same CNS of 0.97 or 0.97%. The CNS error for 3 min ascent dive is 7%.

The total FORTRAN CNS of the reference dive in the predictive mode is .2746 = 27.5%.

The total CNS in the RT mode (1 min clock) is .278 = 27.8%. The error is 1.09% relative to the CNS reference in the predictive mode.

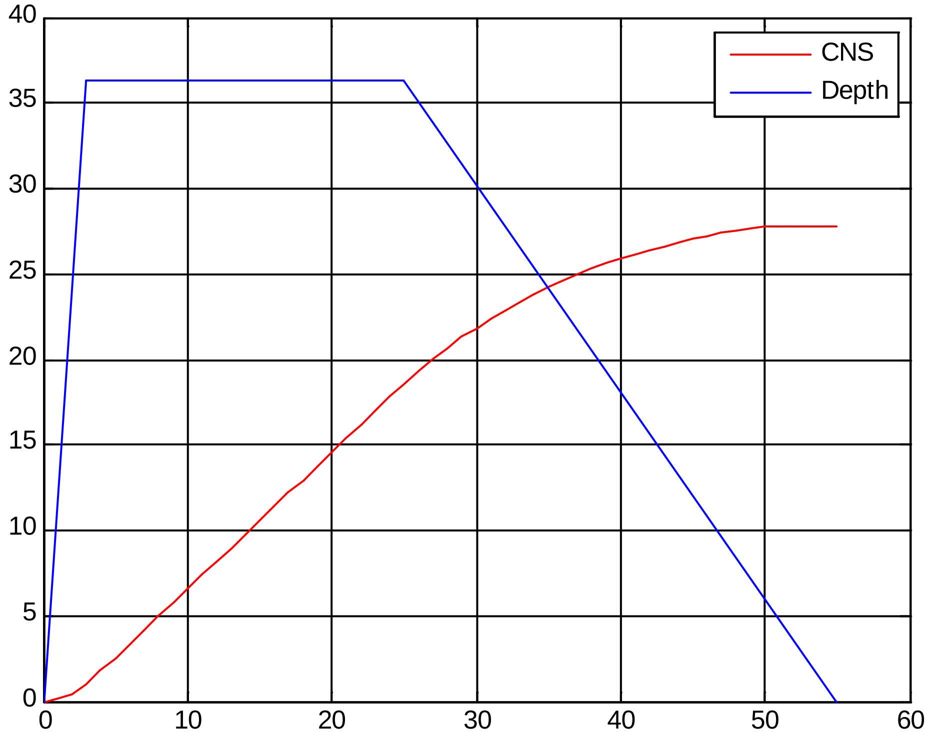


Figure 8. CNS in RT mode, time step is 1 min. max (CNS) = 27.8%, error is 1.09%.

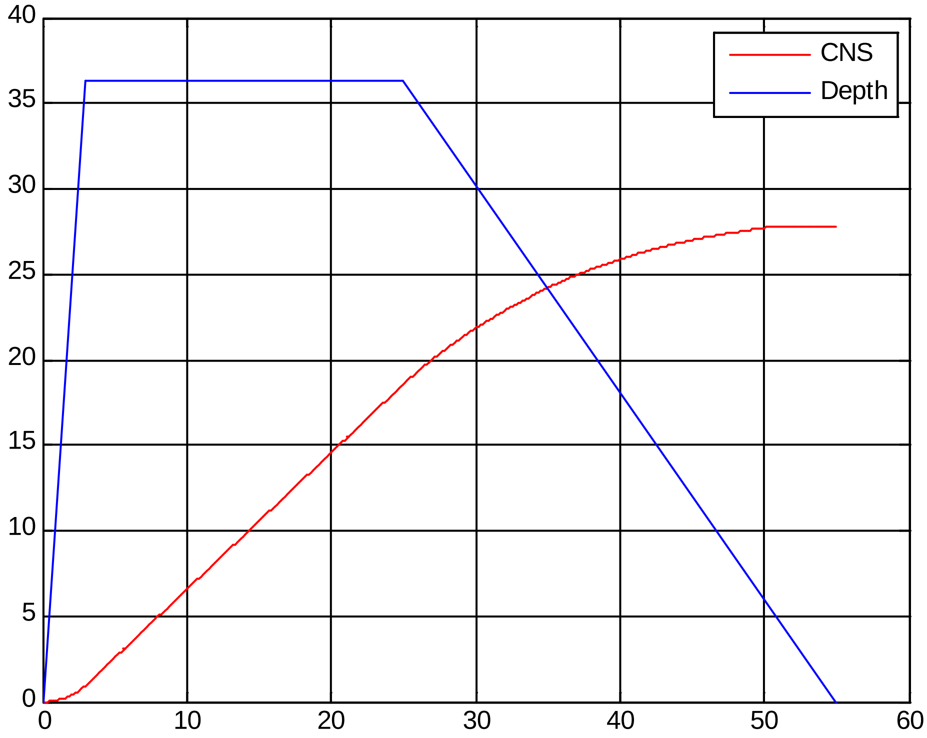


Figure 9. CNS in RT mode, time step is 1s. max (CNS) = 27.74%, error is 0.087%. CNS is not changed when the depth is decreased below 5.62 m (as 10msw/ATTA*(0.5ATA/0.32ATA-1))

8.3 COMPARE RESULTS OF THE FLOATING AND THE INTEGER METHODS

The following data of dive profile [1] is used to compare Erik C. Baker and the suggested method.

1. Descent from 0 fsw to 120 fsw at 40 fsw/min. PPO2 from 0.32 ATA to 1.484
2. Constant depth at 120 fsw for 22 min. PPO2 is 1.484 ATA
3. Long, slow ascent from 120 fsw to 0 fsw at -4 fsw/min PPO2 from 1.484 ATA to 0.32

In the MatLAB format given below, the total CNS of the dive profile is 27.78%. It is the base CNS value. The CNS fractions are calculated for current PPO2 and the corresponding time limit each second. Increasing the time resolution does not change the total CNS value.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CNS = 0;

for t = 1:3300      % 1 s time resolution

    % PPO2
    if t <= 180     % Descent from 0 fsw to 120 fsw at 40 fsw/min
                    % PPO2 from 0.32 ATA to 1.484 ATA
        PPO2 = 0.32 + (1.484-0.32)*t/180;
    end
    if t > 180 && t <= 1500 % Constant depth  at 120 fsw for 22 min
        PPO2 = 1.484;       % in ATA
    end
    if t > 1500     % Ascent from 120 fsw to 0 fsw at 4 fsw/min
                    % PPO2 from 1.484 ATA to 0.32 ATA
        PPO2 = 1.484 - (1.484-0.32)*(t-1500)/1800;
    end

    % Time limit and CNS increment: dCNS
    if PPO2 < 0.5
        dCNS = 0;
    end
    if PPO2 >= 0.5 && PPO2 < 0.6
        CNS_TL = 1800 - 1800*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 0.6 && PPO2 < 0.7
        CNS_TL = 1620 - 1500*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 0.7 && PPO2 < 0.8
        CNS_TL = 1410 - 1200*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 0.8 && PPO2 < 0.9
        CNS_TL = 1170 - 900*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 0.9 && PPO2 < 1.1
        CNS_TL = 900 - 600*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 1.1 && PPO2 < 1.5
        CNS_TL = 570 - 300*PPO2;
        dCNS = 100/(60*CNS_TL);
    end
    if PPO2 >= 1.5 && PPO2 < 1.6
        CNS_TL = 1245 - 750*PPO2;
        dCNS = 100/(60*CNS_TL);
    end

    % total CNS
    CNS = CNS + dCNS;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The total CNS of the suggested method for the compared dive profile is 27.87% for the float calculation and 27% for the calculation of integer numbers as shown in Figure 10.

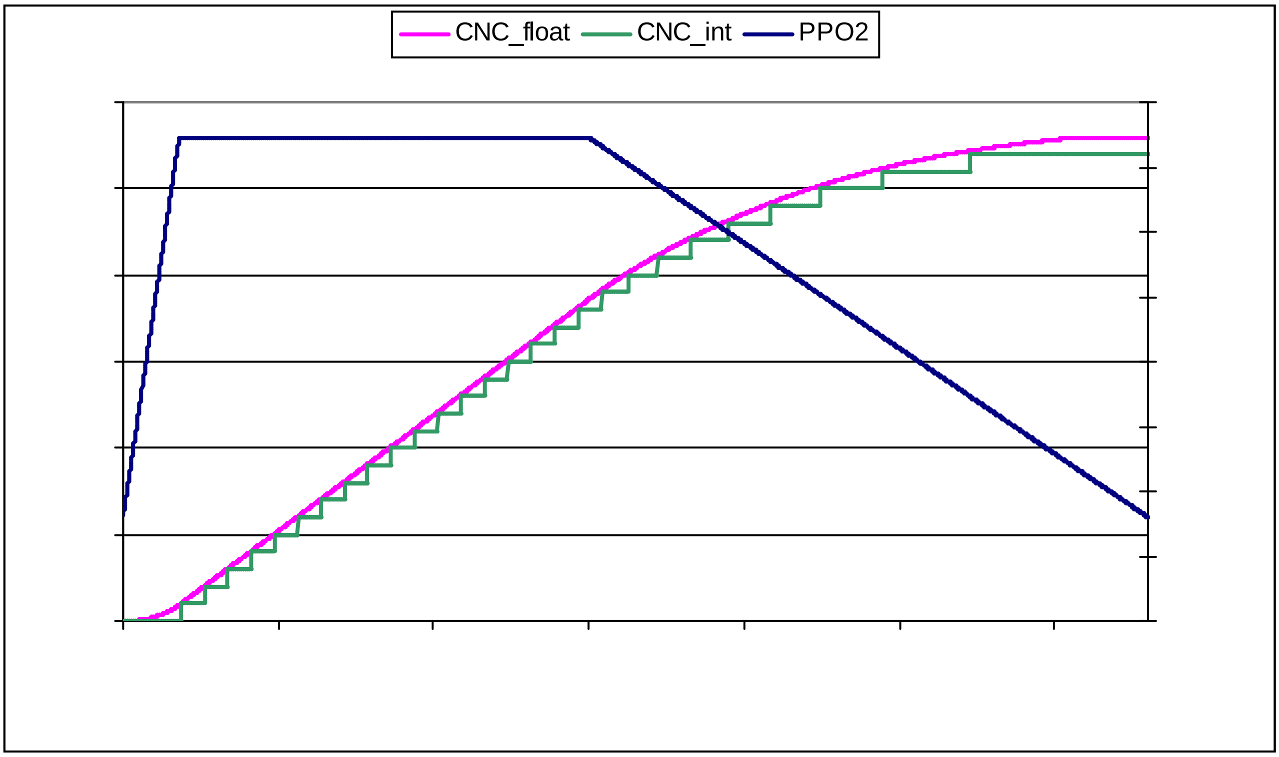


Figure 10. PPO2 profile of Erik C. Baker and the suggested CNS calculated in float and integer data format

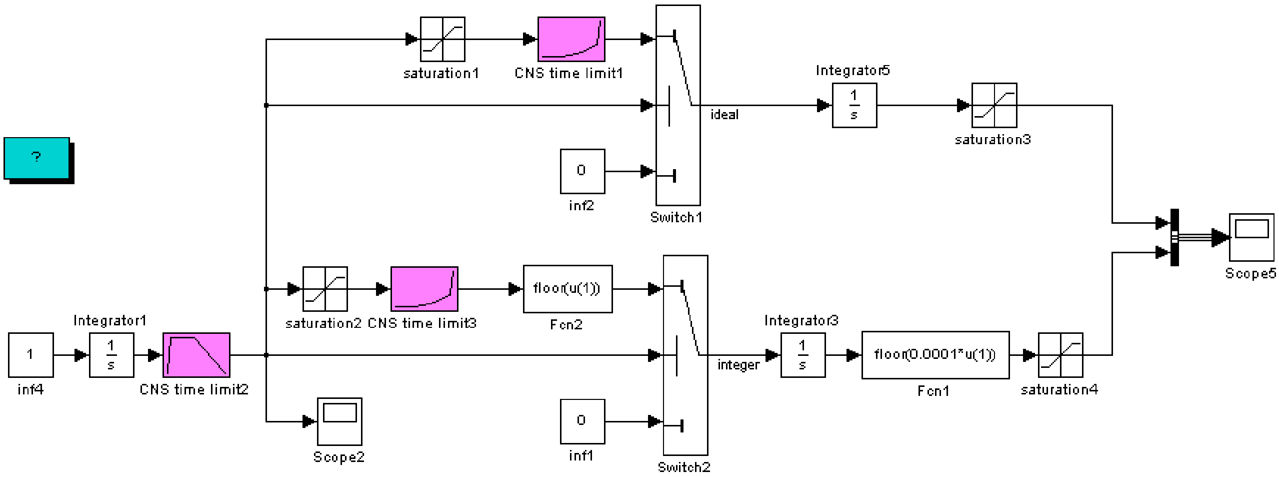


Figure 11. CNS simulink model for calculation with float and integer data

8.4 CNS FOR PPO2 > 1.6 ATA

When PPO2 is more than 1.6 ATA the CNS increment is calculated the same as for PPO2 of 1.6 ATA.

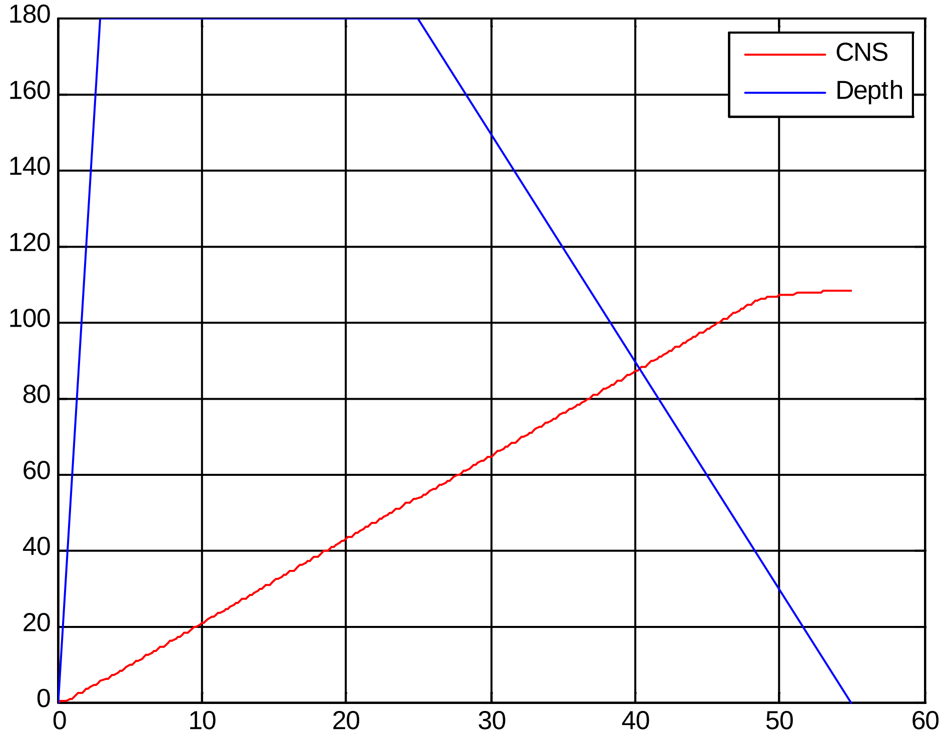


Figure 12. Example of CNS calculation including the PPO2 > 1.6 ATA for EAN32 gas mixture and 3s time step. CNS_max = 108.56%;

ADA code calculated CNS for the same Depth profile, gas mixture and time step shows the max CNS of 109%:

…

```
Run Time:  3294s, Depth:   60cm, NDT remaining:   0min, Stop at   42m, for   1min, TTS:46236s, CNS:  109%
Run Time:  3297s, Depth:   30cm, NDT remaining:   0min, Stop at   42m, for   1min, TTS:46238s, CNS:  109%
```

It corresponds to calculation in MatLAB

The same calculation with the time step increased from 3 s up to 100 s shows max CNS 106%.

8.5 ADA CNS TEST RESULT

The code of the ADA CNS implementation is shown in the APPENDIX of this doc.

8.5.1 Test under the reference dive profile

```
--------------------------------------------------------------------------------
TEST of DECO ALGORITHMs
--------------------------------------------------------------------------------
INPUT TEST NUMBER (1-ZHL16C90, 2-UDM18, 3-Workman65, 4-DCAPMM11F6, 5-DSATRDP, 6-COMEXHe,
10-DCIEM83: 1
INPUT TEST MODE (1-Dive Mode, 2-Planning Mode): 1

INPUT DIVE PROFILE DATA:
Maximum depth    (in m): 36
Bottom_Time    (in min): 22
Ascent_Time    (in min): 30
Time step        (in s): 60
Descent rate (in m/min): 12
Ascent rate  (in m/min): 9

INPUT BREATING MIX (O2 + He + N2 = 100%)
O2 percent   (0 .. 100): 32
He percent   (0 .. 100): 0

PERMITTED LAST STOP DEPTH
Last stop depth (3 .. 6)m : 3

Conservatism (0:10:100)% : 100


TEST RESULTS:
Dive profile.Depth = [0   36   36   0] m;
Dive profile.Time  = [0    3   25   55] min;
O2/He/N2 is 32/0/68%

Conservatism: 100%
Dive starts.
Run Time:    0s, Depth:    0cm, NDT remaining: 99999min, Stop at   0m, for   0min, TTS:    0s, CNS:   0%
Run Time:   60s, Depth: 1200cm, NDT remaining:     6min, Stop at   0m, for   0min, TTS:   80s, CNS:   0%
Run Time:  120s, Depth: 2400cm, NDT remaining:     1min, Stop at   0m, for   0min, TTS:  160s, CNS:   0%
Run Time:  180s, Depth: 3600cm, NDT remaining:     0min, Stop at   3m, for   1min, TTS:  280s, CNS:   1%
Run Time:  240s, Depth: 3600cm, NDT remaining:     0min, Stop at   3m, for   1min, TTS:  280s, CNS:   2%
Run Time:  300s, Depth: 3600cm, NDT remaining:     0min, Stop at   6m, for   1min, TTS:  320s, CNS:   3%
Run Time:  360s, Depth: 3600cm, NDT remaining:     0min, Stop at   9m, for   0min, TTS:  300s, CNS:   3%
Run Time:  420s, Depth: 3600cm, NDT remaining:     0min, Stop at   9m, for   0min, TTS:  300s, CNS:   4%
Run Time:  480s, Depth: 3600cm, NDT remaining:     0min, Stop at  12m, for   1min, TTS:  400s, CNS:   5%
Run Time:  540s, Depth: 3600cm, NDT remaining:     0min, Stop at  12m, for   1min, TTS:  400s, CNS:   6%
Run Time:  600s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  440s, CNS:   6%
Run Time:  660s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  440s, CNS:   7%
Run Time:  720s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  440s, CNS:   8%
Run Time:  780s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  500s, CNS:   9%
Run Time:  840s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  560s, CNS:  10%
Run Time:  900s, Depth: 3600cm, NDT remaining:     0min, Stop at  15m, for   1min, TTS:  620s, CNS:  10%
Run Time:  960s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS:  660s, CNS:  11%
Run Time: 1020s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS:  720s, CNS:  12%
Run Time: 1080s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS:  840s, CNS:  13%
Run Time: 1140s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS:  900s, CNS:  13%
Run Time: 1200s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS:  960s, CNS:  14%
Run Time: 1260s, Depth: 3600cm, NDT remaining:     0min, Stop at  18m, for   0min, TTS: 1020s, CNS:  15%
Run Time:  1320s, Depth:  3600cm, NDT remaining:   0min, Stop at  18m, for   0min, TTS: 1140s, CNS:  16%
Run Time:  1380s, Depth:  3600cm, NDT remaining:   0min, Stop at  18m, for   0min, TTS: 1200s, CNS:  17%
Run Time:  1440s, Depth:  3600cm, NDT remaining:   0min, Stop at  18m, for   0min, TTS: 1320s, CNS:  17%
Run Time:  1500s, Depth:  3600cm, NDT remaining:   0min, Stop at  18m, for   0min, TTS: 1440s, CNS:  18%
Run Time:  1560s, Depth:  3480cm, NDT remaining:   0min, Stop at  18m, for   1min, TTS: 1612s, CNS:  19%
Run Time:  1620s, Depth:  3360cm, NDT remaining:   0min, Stop at  18m, for   1min, TTS: 1664s, CNS:  20%
Run Time:  1680s, Depth:  3240cm, NDT remaining:   0min, Stop at  18m, for   1min, TTS: 1776s, CNS:  20%
Run Time:  1740s, Depth:  3120cm, NDT remaining:   0min, Stop at  15m, for   1min, TTS: 1788s, CNS:  21%
Run Time:  1800s, Depth:  3000cm, NDT remaining:   0min, Stop at  15m, for   1min, TTS: 1840s, CNS:  21%
Run Time:  1860s, Depth:  2880cm, NDT remaining:   0min, Stop at  15m, for   1min, TTS: 1892s, CNS:  22%
Run Time:  1920s, Depth:  2760cm, NDT remaining:   0min, Stop at  15m, for   1min, TTS: 1944s, CNS:  22%
Run Time:  1980s, Depth:  2640cm, NDT remaining:   0min, Stop at  15m, for   1min, TTS: 1996s, CNS:  23%
Run Time:  2040s, Depth:  2520cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 2008s, CNS:  23%
Run Time:  2100s, Depth:  2400cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 2000s, CNS:  24%
Run Time:  2160s, Depth:  2280cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 1992s, CNS:  24%
Run Time:  2220s, Depth:  2160cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 2044s, CNS:  24%
Run Time:  2280s, Depth:  2040cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 2096s, CNS:  25%
Run Time:  2340s, Depth:  1920cm, NDT remaining:   0min, Stop at  12m, for   1min, TTS: 2028s, CNS:  25%
Run Time:  2400s, Depth:  1800cm, NDT remaining:   0min, Stop at   9m, for  2min, TTS: 1980s, CNS:  25%
Run Time:  2460s, Depth:  1680cm, NDT remaining:   0min, Stop at   9m, for  2min, TTS: 2032s, CNS:  26%
Run Time:  2520s, Depth:  1560cm, NDT remaining:   0min, Stop at   9m, for  2min, TTS: 2024s, CNS:  26%
Run Time:  2580s, Depth:  1440cm, NDT remaining:   0min, Stop at   9m, for  1min, TTS: 2016s, CNS:  26%
Run Time:  2640s, Depth:  1320cm, NDT remaining:   0min, Stop at   9m, for  1min, TTS: 1948s, CNS:  26%
Run Time:  2700s, Depth:  1200cm, NDT remaining:   0min, Stop at   9m, for  1min, TTS: 1940s, CNS:  26%
Run Time:  2760s, Depth:  1080cm, NDT remaining:   0min, Stop at   6m, for  9min, TTS: 1892s, CNS:  27%
Run Time:  2820s, Depth:   960cm, NDT remaining:   0min, Stop at   6m, for  8min, TTS: 1884s, CNS:  27%
Run Time:  2880s, Depth:   840cm, NDT remaining:   0min, Stop at   6m, for  7min, TTS: 1816s, CNS:  27%
Run Time:  2940s, Depth:   720cm, NDT remaining:   0min, Stop at   6m, for  6min, TTS: 1748s, CNS:  27%
Run Time:  3000s, Depth:   600cm, NDT remaining:   0min, Stop at   6m, for  5min, TTS: 1680s, CNS:  27%
Run Time:  3060s, Depth:   480cm, NDT remaining:   0min, Stop at   6m, for  4min, TTS: 1628s, CNS:  27%
Run Time:  3120s, Depth:   360cm, NDT remaining:   0min, Stop at   6m, for  3min, TTS: 1576s, CNS:  27%
Run Time:  3180s, Depth:   240cm, NDT remaining:   0min, Stop at   3m, for 24min, TTS: 1444s, CNS:  27%
Run Time:  3240s, Depth:   120cm, NDT remaining:   0min, Stop at   3m, for 23min, TTS: 1392s, CNS:  27%
Deco end
```

The CNS for the depth profile Depth_Profile_m = (0 36 36 0) in msw and Time_prof_min = (1 3 25 55) in min corresponds to the test resuls of CNS in MatLAB for the same dive profile and the gas mix of EAN32

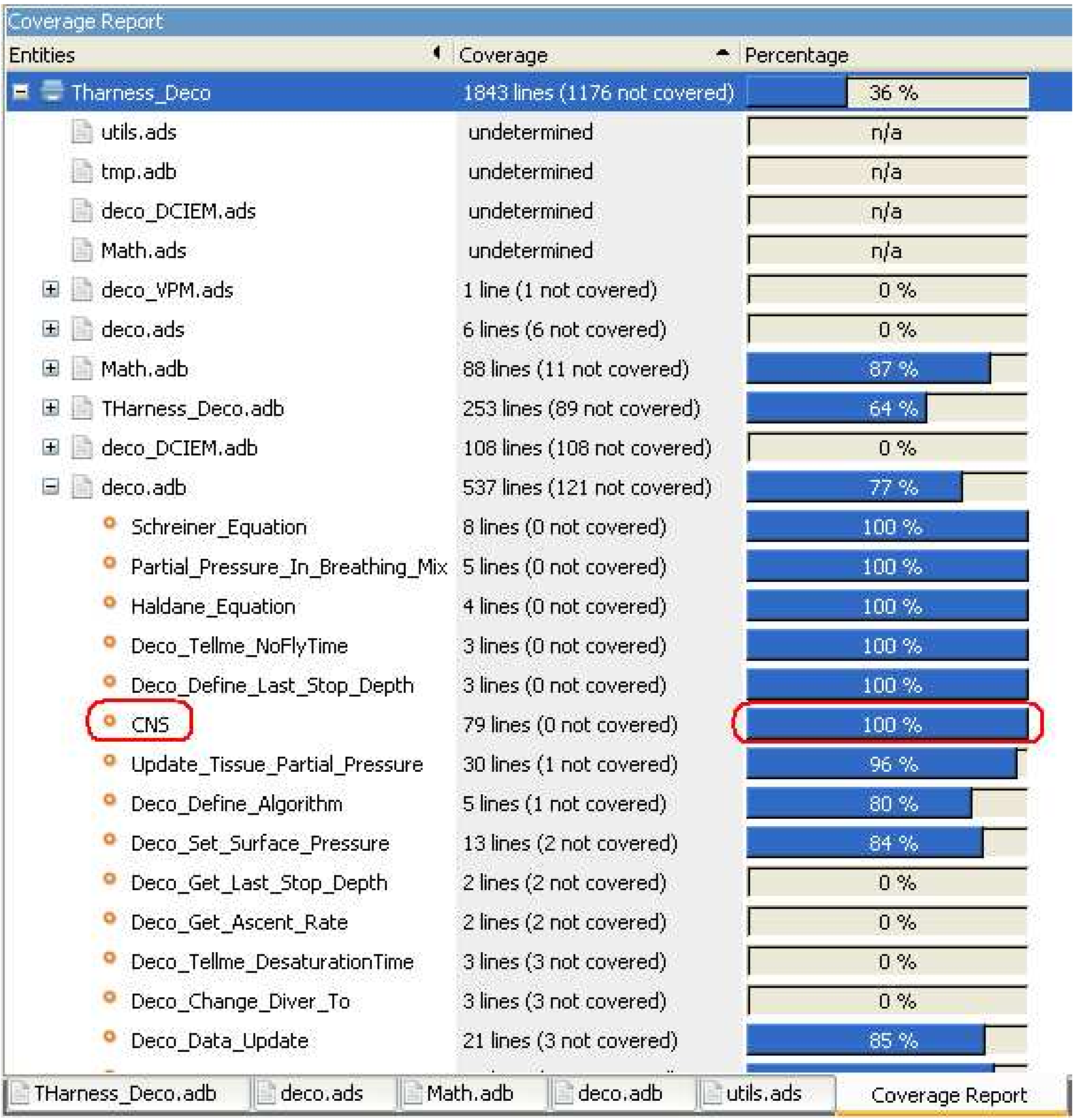
8.5.2 ADA Test Harness Example

The following dive profile and gas mix provides the 92% coverage of ADA CNS code, with 100% achieved by this example then re-run using pure O2.

```
--------------------------------------------------------------------------------
TEST of DECO ALGORITHMs
--------------------------------------------------------------------------------
INPUT TEST NUMBER (1-ZHL16C90, 2-UDM18, 3-Workman65, 4-DCAPMM11F6, 5-DSATRDP, 6-COMEXHe, 7-
DCIEM2010, 8-VPM, 9-VPM-B, 10-DCIEM83: 1
INPUT TEST MODE (1-Dive Mode, 2-Planning Mode): 1

INPUT DIVE PROFILE DATA:
Maximum depth    (in m): 40  -- then 100m in second run
Bottom_Time     (in min): 30
Ascent_Time     (in min): 3
Time step        (in s): 60
```

```
Descent rate (in m/min): 12
Ascent rate  (in m/min): 9

INPUT BREATING MIX (O2 + He + N2 = 100%)
O2 percent   (0 .. 100): 32  -- then pure O2 in second run
He percent   (0 .. 100): 0

PERMITTED LAST STOP DEPTH
Last stop depth (3 .. 6)m : 3

Conservatism (0:10:100)% : 100


TEST RESULTS:
Dive profile.Depth = [0   40   40  0] m;
Dive profile.Time  = [0    3   33  36] min;
O2/He/N2 is 32/0/68%

Conservatism: 100%
Dive starts.
Run Time:     0s, Depth:     0cm, NDT remaining: 99999min, Stop at    0m, for    0min, TTS:    0s, CNS:    0%
Run Time:    60s, Depth:  1200cm, NDT remaining:     6min, Stop at    0m, for    0min, TTS:   80s, CNS:    0%
Run Time:   120s, Depth:  2400cm, NDT remaining:     1min, Stop at    0m, for    0min, TTS:  160s, CNS:    0%
Run Time:   180s, Depth:  3600cm, NDT remaining:     0min, Stop at    3m, for    1min, TTS:  280s, CNS:    1%
Run Time:   240s, Depth:  4000cm, NDT remaining:     0min, Stop at    3m, for    1min, TTS:  307s, CNS:    2%
Run Time:   300s, Depth:  4000cm, NDT remaining:     0min, Stop at    6m, for    1min, TTS:  347s, CNS:    4%
Run Time:   360s, Depth:  4000cm, NDT remaining:     0min, Stop at    9m, for    1min, TTS:  387s, CNS:    7%
Run Time:   420s, Depth:  4000cm, NDT remaining:     0min, Stop at   12m, for    1min, TTS:  427s, CNS:    9%
Run Time:   480s, Depth:  4000cm, NDT remaining:     0min, Stop at   12m, for    1min, TTS:  427s, CNS:   11%
Run Time:   540s, Depth:  4000cm, NDT remaining:     0min, Stop at   15m, for    1min, TTS:  467s, CNS:   13%
Run Time:   600s, Depth:  4000cm, NDT remaining:     0min, Stop at   15m, for    1min, TTS:  467s, CNS:   15%
Run Time:   660s, Depth:  4000cm, NDT remaining:     0min, Stop at   15m, for    1min, TTS:  527s, CNS:   18%
Run Time:   720s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  627s, CNS:   20%
Run Time:   780s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  687s, CNS:   22%
Run Time:   840s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  747s, CNS:   24%
Run Time:   900s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  867s, CNS:   27%
Run Time:   960s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  927s, CNS:   29%
Run Time:  1020s, Depth:  4000cm, NDT remaining:     0min, Stop at   18m, for    1min, TTS:  987s, CNS:   31%
Run Time:  1080s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1147s, CNS:   33%
Run Time:  1140s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1267s, CNS:   35%
Run Time:  1200s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1327s, CNS:   38%
Run Time:  1260s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1447s, CNS:   40%
Run Time:  1320s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1567s, CNS:   42%
Run Time:  1380s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1747s, CNS:   44%
Run Time:  1440s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 1867s, CNS:   47%
Run Time:  1500s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2047s, CNS:   49%
Run Time:  1560s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2167s, CNS:   51%
Run Time:  1620s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2287s, CNS:   53%
Run Time:  1680s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2347s, CNS:   55%
Run Time:  1740s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2467s, CNS:   58%
Run Time:  1800s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2587s, CNS:   60%
Run Time:  1860s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2767s, CNS:   62%
Run Time:  1920s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 2947s, CNS:   64%
Run Time:  1980s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 3067s, CNS:   67%
Run Time:  2040s, Depth:  4000cm, NDT remaining:     0min, Stop at   21m, for    1min, TTS: 3187s, CNS:   69%
Run Time:  2100s, Depth:  2667cm, NDT remaining:    0min, Stop at    18m, for     1min, TTS: 3058s, CNS:    70%
Run Time:  2160s, Depth:  1333cm, NDT remaining:    0min, Stop at    12m, for     4min, TTS: 2889s, CNS:    70%
Run Time:  2220s, Depth:     0cm, NDT remaining:    0min, Stop at     9m, for     7min, TTS: 2700s, CNS:   70%
Deco end
```

8.6 TEST COVERAGE



Entities	Coverage	Percentage
Tharness_Deco	1843 lines (1176 not covered)	36 %
utils.ads	undetermined	n/a
tmp.adb	undetermined	n/a
deco_DCIEM.ads	undetermined	n/a
Math.ads	undetermined	n/a
deco_VPM.ads	1 line (1 not covered)	0 %
deco.ads	6 lines (6 not covered)	0 %
Math.adb	88 lines (11 not covered)	87 %
THarness_Deco.adb	253 lines (89 not covered)	64 %
deco_DCIEM.adb	108 lines (108 not covered)	0 %
deco.adb	537 lines (121 not covered)	77 %
Schreiner_Equation	8 lines (0 not covered)	100 %
Partial_Pressure_In_Breathing_Mix	5 lines (0 not covered)	100 %
Haldane_Equation	4 lines (0 not covered)	100 %
Deco_Tellme_NoFlyTime	3 lines (0 not covered)	100 %
Deco_Define_Last_Stop_Depth	3 lines (0 not covered)	100 %
CNS	79 lines (0 not covered)	100 %
Update_Tissue_Partial_Pressure	30 lines (1 not covered)	96 %
Deco_Define_Algorithm	5 lines (1 not covered)	80 %
Deco_Set_Surface_Pressure	13 lines (2 not covered)	84 %
Deco_Get_Last_Stop_Depth	2 lines (2 not covered)	0 %
Deco_Get_Ascent_Rate	2 lines (2 not covered)	0 %
Deco_Tellme_DesaturationTime	3 lines (3 not covered)	0 %
Deco_Change_Diver_To	3 lines (3 not covered)	0 %
Deco_Data_Update	21 lines (3 not covered)	85 %

Figure 13. The Test Coverage of the CNS algorithm is 100%: this is the only module intended to be covered by the test harness used above – the coverage of other modules is incidental – they have their own test harnesses that provide 100% cover.

The test harness for the CNS module has 100% Statement, Branch and Bound coverage.

The test harness covers PPO2 levels from 0.0 to 10.0: a predicate is used in the enveloping routine to reject PPO2 values outside this range.

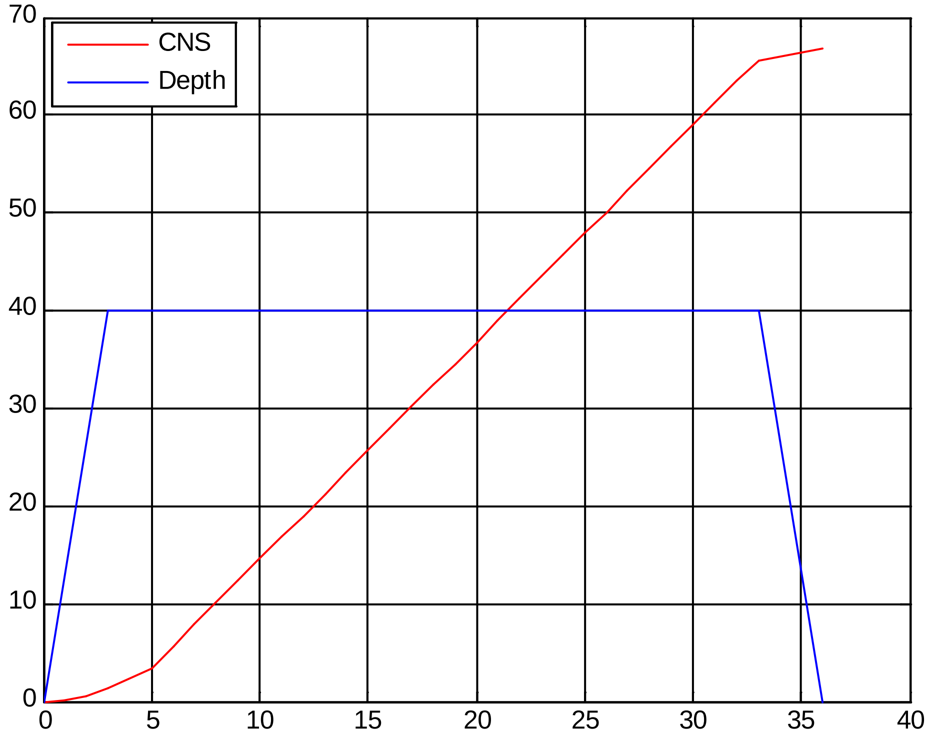


Figure 14. The MatLAB CNS against time. The dive profile and gas mix used for 100% harness of ADA code is applied to the CNS MatLAB code. CNS_max = 66.7707%. It corresponds to the ADA max CNS of 77% (CNS resolution increment is 1%).

8.7 Confidence of Correctness

The CNS implementation in SPARK Ada comprises just 140 lines of code.

Software development processes used by Deep Life are monitored using the Carnegie Mellon PSP and TSP methods, to achieve less than one defect per 10KLOC of code. This is also the intrinsic defect rate of SPARK programs without PSP. Therefore the probability of an implementation defect in the Deep Life DCIEM implementation is less than 1.4%.

These are upper bounds, as it is likely the case that as each of SPARK Ada and PSP on their own achieves a 0.1 / KLOC defect rate, then the deliverable defect rates can be 1 defect per 100 KLOC or better – ideally the two metrics would combine as the product but in safety engineering such optimism would be unacceptable.

9 Conclusion

1. Test results of the MatLAB CNS model corresponds to the FORTRAN reference results.
2. Test results of the ADA CNS code corresponds to the MatLAB model and the FORTRAN reference results.
3. CNS function is calculated correctly for PPO2 up to 1.6 ATA. CNS is undefined above 1.6 ATA in the reference material but appropriate functions are proposed and modelled.

4. The software module the CNS reporting is combined with generates warnings and alarms in accord with EN 14143:2003 and EN 14143:2013 in the event the PPO2 exceeding 1.6 ATA.
5. During integration testing, both the CNS computation and the associated warnings and alarms are tested and validated.

10 REFERENCES

The use of academic references has been avoided but some references are important to allow this work to be substantiated, and to form a traceable scientific basis for the verification and determinations developed herein.

Where a choice of references exist, in general the reference readily available on the internet is chosen over those available only via a traditional library service. The Rubicon library has been of special value in making papers available online: RRR refers to the Rubicon Repository.

References:

1. Oxygen Toxicity Calculations, Erik C. Baker, P.E. www.dmscuba.com/Oxygen_Toxicity_Calculations.pdf
2. Decompression Practice, Robert W Hamilton and Ed Thalmann http://www.intl.elsevierhealth.com/e-books/pdf/317.pdf
3. Gas Mixing and Dive Planning Spreadsheets and Charts http://mywebpages.comcast.net/jeff.hunter/mixcharts.html
4. OXYGEN EXPOSURE MANAGEMENT, Richard D. Vann http://bluenine.tv/johan/rebreather/files/OxygenExposure.html
5. PCO2 Threshold for CNS oxygen toxicity in rats in the low range of hyperbaric PPO2, R. Arieli et al., J. Appl. Physiologgy 01:1582-1587, 2001. http://jap.physiology.org/cgi/reprint/91/4/1582.pdf
6. Effects of nitrogen and helium on CNS toxicity in the rat, R. Arieli et. Al., J. Appl. Physiology 98:144-150, 2005
7. DAN Diving Medicine Articles. OXTOX: if you dive nitrox you should know about OXTOX, E. Thalmann. DAN internet site.
8. A Model for Predicting Central Nervous System Toxicity from Hyperbaric Oxygen Exposures in Humans, A. Harabin, S. Survanshi and L. Homer, Toxicology and Applied Pharmacology, V132(1), May 1995, pp 19.26. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=7747281&dopt=Abstract
9. Central Nervous System oxygen toxicity in closed circuit scuba divers II, F. Butler and E. Thalmann, Undersea Biomed Res. V13(2), 1986 Jun; pp193-223 http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=3727183&query_hl=1&itool=pubmed_docsum
10. Factors Affecting CNS Oxygen Toxicity in Humans. Natoli and Vann. 1996 MS Thesis. Rubicon Reposity Hyperlink ID: 21
11. CNS (Central Nervous System) Oxygen Toxicity in Closed-Circuit SCUBA Divers. Butler and Thalmann. NEDU 11-84; 1984. RRR ID: 3378
12. Central Nervous System Oxygen Toxicity in Closed-Circuit SCUBA Divers II. Butler FK and Thalmann ED. Undersea Biomedical Research 13(12); 193-224; June 1986. RRR ID: 3045
13. Central Nervous System Oxygen Toxicity in Closed-Circuit SCUBA Divers III. Butler FK. NEDU Report 5-86, May 1986. RRR ID: 4511

14. Screening for Oxygen Intolerance in U.S. Navy Divers. Butler FK and Knafelc ME. Undersea Biomedical Research 13(1); 91-98; March, 1986. RRR ID: 3046
15. Screening for Oxygen Tolerance in U.S. Navy Combat Swimmers. Walters K, Gould M, Bacharach E, Butler F. Undersea Hyper Med 2000; 27: 21-26. RRR ID: 2358
16. A test for variations in individual sensitivity to hyperbaric oxygen toxicity. Harabin, Survanshi, and Homer. Undersea Hyperb Med. 1994 Dec;21(4):403-12. RRR ID: 2154
17. CNS oxygen toxicity in oxygen-inert gas mixtures. Bitterman, Laor, and Melamed. Undersea Biomed Res. 1987 Nov;14(6):477-83. RRR ID: 2445
18. Visual Reaction Time Performance Preceding CNS Oxygen Toxicity. Curley MD and Butler FK. Undersea Biomedical Research 14(4); 301-310; July, 1987. RRR ID: 3086
19. Oxygen Toxicity – Pulmonary. Two Consecutive Five-Day Weeks of Daily Four-Hour Dives with Oxygen Partial Pressure 1.4 ATM Shykoff. 2006 NEDU Report. RRR ID: 3493
20. Extension of oxygen tolerance by interrupted exposure. Clark. Undersea Hyperb Med. 2004 Summer;31(2):195-8. RRR ID: 4009
21. Pulmonary effects of submerged oxygen breathing: 4-, 6-, and 8-hour dives at 140 kPa. Shykoff Undersea Hyperb Med. 2005 Sep-Oct;32(5):351-61. RRR ID: 4031
22. Pulmonary Oxygen Tolerance in Man and Derivation of Pulmonary Oxygen Tolerance Curves. Clark and Lambertsen. 1970 PhD Thesis. RRR ID: 3863
23. Comprehensive Performance Limits for Divers' underwater breathing gear: consequences of adopting diver-focused limits. D. Warkander. 2007 NEDU Report number TR 07-02.

Appendix

ADA CNS code Revision 20th May 2013 with Square Fn for PPO2 > 1.6 ATM

The name of variables is taken from the Eric Baker FORTRAN model to ease comparison: this is not DL's normal procedure, but given the brevity of the algorithm and the documentation was considered the best naming method in this instance.

Note the SPARK attributes have been removed from the Ada listing below, for reasons of confidentiality and licensing.

```
package body Deco is
…
  CNS_crnt                    : Float    := 0.0;
…

  PO2LO   : Array_7 := (    0.5,     0.6,     0.7,    0.8,    0.9,     1.1,    1.5);  -- low end of PO2 range
  PO2HI   : Array_7 := (    0.6,     0.7,     0.8,    0.9,    1.1,     1.5,    1.6);  --  high end of PO2 range
  LIMSLP   : Array_7 := (-1800.0, -1500.0, -1200.0, -900.0, -600.0,   -300.0, -750.0);  --  slope coefficient for PO2
range
  LIMINT  : Array_7 := ( 1800.0,  1620.0,  1410.0, 1170.0,   900.0,  570.0, 1245.0);  --  intercept coefficient for PO2
range
  CNS_tmp : Array_7 := (    0.0, others => 0.0);
  PO2O    : Array_7 := (    0.0, others => 0.0);
  OTIME   : Array_7 := (    0.0, others => 0.0);
  SEGPO2  : Array_7 := (    0.0, others => 0.0);
```

10.1.1 Reset CNS

```
procedure Deco_Set_Surface_Pressure (Last_Surface_Pressure : mbar_abs) is
…
    CNS_crnt := 0.0;
…
  end Deco_Set_Surface_Pressure;
```

10.1.2 Running CNS in RT mode

```
function Deco_Data_Update (OC_Mode : Boolean; AGas_percent : Gas; PPO2_mbar : mbar_abs;
                           Current_Depth_cm : Depth_cm; Current_Time_s : HHMMSS) return Deco_status is
…
    CNS_inc := CNS (Depth_Old_m, Depth_m, AGas_current_percent, Sample_Time_min);
    CNS_crnt := CNS_crnt + CNS_inc;

    Deco_Status_List.CNS := Percents (CNS_crnt * 100.0);
…
  end Deco_Data_Update;
```

10.1.3 CNS.adb

```
function CNS

--  Each clock calculates CNS
    (Depth_Old_m      : Float;
     Depth_m          : Float;
     Gas_percent      : Gas;
     Sample_Time_min  : Float) return Float is
```

```
   CNS_Clock_time  : Float;
   CNS_inc_frac : Float;
   FO2     : Float;
   FPATA   : Float;
   IPATA   : Float;
   MAXATA  : Float;
   MINATA  : Float;
   MAXPO2  : Float;
   MINPO2  : Float;
   LOWPO2  : Float;
   O2TIME  : Float;
   PO2F    : Float;
   TLIMI   : Float;
   MK      : Float;

begin
   CNS_inc_frac := 0.0;
   FO2 := Float (Gas_percent.O2) / 100.0;

   for I in 1 .. 7 loop
      CNS_tmp (I) := 0.0;
      PO2O (I) := 0.0;
      OTIME (I) := 0.0;
      SEGPO2 (I) := 0.0;
   end loop;

   IPATA := (Depth_Old_m + 10.0) / 10.0;
   FPATA := (Depth_m + 10.0) / 10.0;
   if IPATA > FPATA then
      MAXATA := IPATA;
   else
      MAXATA := FPATA;
   end if;
   MAXPO2 := MAXATA * FO2;
   if MAXPO2 > 1.6 then
      CNS_Clock_time := 45.0 / (((MAXPO2 + 0.1) - 1.6) * ((MAXPO2 + 0.1) - 1.6) * 100.0);
      CNS_inc_frac := Sample_Time_min / CNS_Clock_time;
   else
      if IPATA < FPATA then
         MINATA := IPATA;
      else
         MINATA := FPATA;
      end if;
      MINPO2 := MINATA * FO2;
      if MAXPO2 > 0.5 then
         if MINPO2 < 0.5 then
            LOWPO2 := 0.5;
         else
            LOWPO2 := MINPO2;
         end if;
         O2TIME := Sample_Time_min * (MAXPO2 - LOWPO2) / (MAXPO2 - MINPO2);
         for I in 1 .. 7 loop
            if ((MAXPO2 > PO2LO(I)) and then (LOWPO2 <= PO2HI(I))) then
               if ((MAXPO2 >= PO2HI(I)) and then (LOWPO2 < PO2LO(I))) then
```

```
          if (Depth_Old_m > Depth_m) then
            PO2O(I) := PO2HI(I);
            PO2F := PO2LO(I);
          else
            PO2O(I) := PO2LO(I);
            PO2F := PO2HI(I);
          end if;
          SEGPO2(I) := PO2F - PO2O(I);
        elsif ((MAXPO2 < PO2HI(I)) and then  (LOWPO2 <= PO2LO(I))) then
          if (Depth_Old_m > Depth_m) then
            PO2O(I) := MAXPO2;
            PO2F := PO2LO(I);
          else
            PO2O(I) := PO2LO(I);
            PO2F := MAXPO2;
          end if;
          SEGPO2(I) := PO2F - PO2O(I);
        elsif ((LOWPO2 > PO2LO(I)) and then (MAXPO2 >= PO2HI(I))) then
          if (Depth_Old_m > Depth_m) then
            PO2O(I) := PO2HI(I);
            PO2F := LOWPO2;
          else
            PO2O(I) := LOWPO2;
            PO2F := PO2HI(I);
          end if;
          SEGPO2(I) := PO2F - PO2O(I);
        else
          if (Depth_Old_m > Depth_m) then
            PO2O(I) := MAXPO2;
            PO2F := LOWPO2;
          else
            PO2O(I) := LOWPO2;
            PO2F := MAXPO2;
          end if;
          SEGPO2(I) := PO2F - PO2O(I);
        end if;
        OTIME(I) := O2TIME * abs (SEGPO2(I))/(MAXPO2 - LOWPO2);
      else
        OTIME(I) := 0.0;
      end if;
    end loop;
    for I in 1 .. 7 loop
      if OTIME(I) = 0.0 then
        CNS_tmp(I) := 0.0;
      else
        TLIMI := LIMSLP(I)*PO2O(I) + LIMINT(I);
        MK := LIMSLP(I)*(SEGPO2(I)/OTIME(I));
        CNS_tmp(I) := 1.0 / MK * (Math.DL_Log (abs (TLIMI + MK * OTIME(I))) - Math.DL_Log (abs (TLIMI)));
      end if;
    end loop;
    for I in 1 .. 7 loop
      CNS_inc_frac := CNS_inc_frac + CNS_tmp (I);
    end loop;
  end if;
end if;
```

```
    return CNS_inc_frac;
  end CNS;
```

10.1.4 CNS.ads

```
function CNS
  --  Each TTA clock calculates CNS
    (Depth_Old_m      : Float;
     Depth_m          : Float;
     Gas_percent      : Gas;
     Sample_Time_min  : Float) return Float;
```