Instruction Manual for Certification Workbook

The workbook is called "Certification Analysis.xlsx." This version is to be used after a 12-test leak detection system certification project in a single tank (the *certification tank*).

1. Entering Data

Data from a hypothetical evaluation are in the table below. The calibration was designed to be run at four target leak rates, in this case (0.0, 0.3,0.6, and 0.9 gph). It was not possible to hit these values exactly and the actual induced leak rates (ILR) as measured, for example, by stop-watch and graduated cylinder or digital timer and load cell, are the independent or x-variable in the calibration analysis. Measured leak rates (MLR) reported by the detection system under evaluation are the dependent or y-variable in the analysis.

	Leak Rate		Measured Leak Rate
Test No	Intended	ILR (x)	MLR (y)
1	0.6	0.450	0.496
2	0.0	0.000	0.200
3	0.9	0.893	0.973
4	0.3	0.233	0.519
5	0.0	0.000	0.040
6	0.9	0.922	0.960
7	0.6	0.636	0.700
8	0.3	0.304	0.360
9	0.3	0.200	-0.130
10	0.0	0.000	-0.100
11	0.9	0.749	1.150
12	0.6	0.560	0.753

Tip:

Keep an unpopulated master copy of the workbook in a master directory. Open that copy and save it under a different name in the directory for your current evaluation project, then type in the ILR and MLR values and resave.

Enter tank specs and measured and induced leak rates in the workbook:

1. Read the Instruction manual that came with this worksheet.
"Instruction Manual for Certification Analysis Workbook.pdf"

2. Enter Calibration Tank Specs	
Product Surface Area (sq ft)	14,039
Test Duration (hrs)	72

3. Enter Evaluation Data		
Test No	ILR (x)	MLR (y)
1	0.450	0.496
2	0.000	0.200
3	0.893	0.973
4	0.233	0.519
5	0.000	0.040
6	0.922	0.960
7	0.636	0.700
8	0.304	0.360
9	0.200	-0.130
10	0.000	-0.100
11	0.749	1.150
12	0.560	0.753
Means:	**0.412**	**0.493**

Figure 1. Data entry section of Certification Worksheet.

2. The Calibration Curve

The workbook computes a bivariate linear regression with 5% and 95% prediction limits (Figure 2, next page). The critical value of the measured leak rate (L_C) is the y-coordinate of the 95% prediction limit at an induced leak rate of x = 0 gph. The probability that only 5% of future measured leak rates of a tight tank will exceed L_C. The lower detection limit (L_D) of a true leak rate is the x-coordinate of where the 5% prediction limit reaches the critical value L_C.

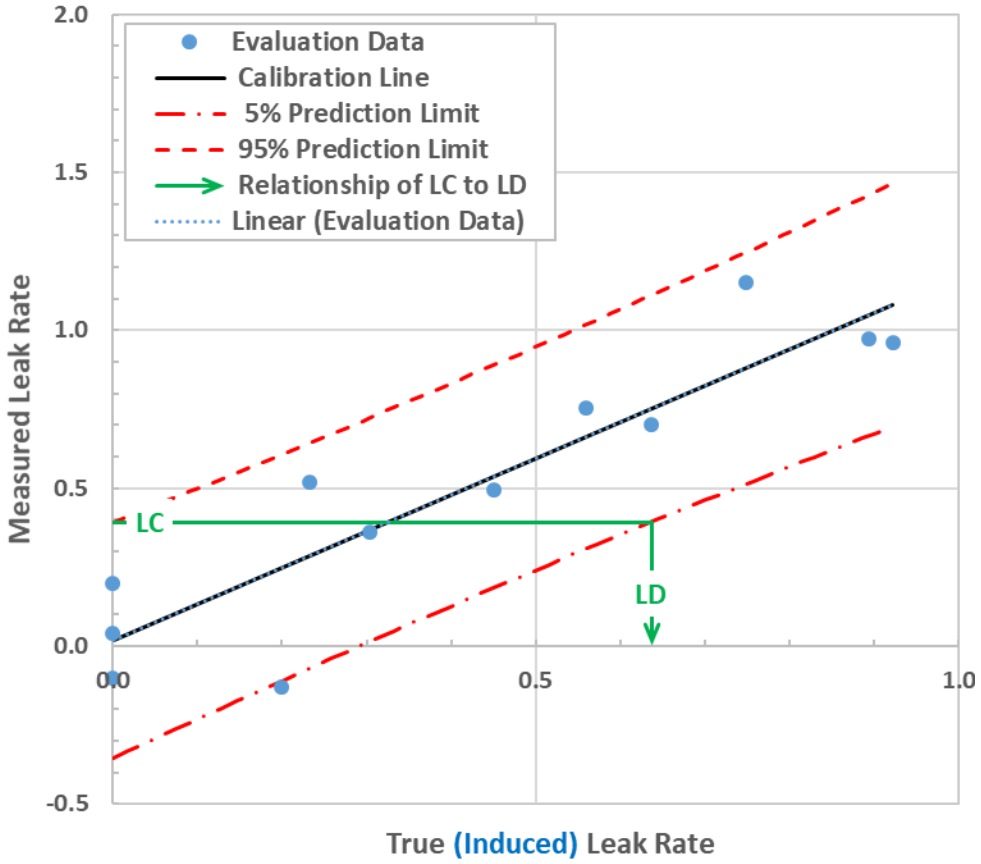

Figure 2. Graph of Linear Regression with Prediction Limits

3. Action and Detection Limits

Estimated values of LC and LD are reported in box 3 on the first page of the workbook

3. Results: Action and Detection Limits	
Critical Value (LC)	0.393 gph
Minimum Detectable Value (LD)	0.637 gph

If these values (LC = 0.509 and LD = 0.99) are used in routine leak testing of the evaluation tank, then the false alarm rate will be 5% and at least 95% of leaks greater than 0.994 gph, will be detected. In statistical theory the same claim can be made about any tank that is *fungible*[1] with the evaluation tank.

4. Scaling Equations

In practice the detection system vendor may choose to provide *scaling equations* to adjust LC and LD for use in monitored tanks with different dimensions than the calibration tank. For

[1] Practically interchangeable, in particular, having the same dimensions.

mass-based leak detection systems a plausible, ad-hoc adjustment is to multiply LD and LD by the surface area of the monitored tank divided by the surface area of the calibration tank.

Suggested scaling equations are reported in Box 4 on the first page of the workbox,

4. Scaling to larger/smaller tank
LC = 0.019 + 0.374 x (SA.new/14,039) x SQRT(72/Dur.new)
LD = 0.637 x (SA.new/14,039) x SQRT(72/Dur.new)

References

[1] Hubaux, A., and Vos, G., "Decision and Detection Limits for Linear Calibration Curves," Analytical Chemistry, Vol. 42, No. 8, July 1970. https://www.dropbox.com/s/f2b9j3drt9ne8jb/Decision%20and%20Detection%20Limits%20for%20Linear%20Calibration%20Curves%201970%20Hubaux%20%26%20Vos.pdf?dl=0

[2] Currie, Lloyd A., "Nomenclature in Evaluation of Analytical Methods Including Detection and Quantification Capabilities," Pure &Appl. Chem., Vol. 67, No. 10, pp. 1699-1723, 1995. https://www.dropbox.com/s/cgn0fltqcqykwvq/Nomenclature%20Currie.pdf?dl=0

[3] Ken Wilcox Associates, "Addendum to the Evaluation of the LRDP-24 on Bulk Field-Constructed Tanks, Final Report," August 7, 2001. https://www.dropbox.com/s/zs1mobm3iif0fys/KWA_LRDP_Protocol_Navy-Vista-VPSI_LRDP_KWA%20Report_NFESC-Vista_Red_Hill_08-15-01_041814.pdf?dl=0

Annex B: Regression Calibration

This section is provided as documentation, but it is not necessary to read or understand it to use the Workbook.

1. Linear Regression

For dependent variable y and independent variable x, a multiple linear regression expresses y as random noise plus the weighted sum of two or more functions of x, called *basis functions*. For bivariate linear regressions the basis functions are the constant: $b_1(x) = 1$, and x itself, $b_2(x) = x$. The weights (regression coefficients) are symbolized as subscripted betas, so

$$y = \beta_1 \cdot b_1(x) + \beta_2 \cdot b_2(x) + noise$$

is ordinary straight-line regression (bivariate linear regression):

$$y = \beta_1 + \beta_2 \cdot x + noise.$$

The matrix of values of the basis functions is called the *design matrix*, traditionally symbolized by bold capital **X**. Each row is data from one test run and each column is a basis function. The vector of y values is symbolized by bold lower-case **y**, and the vector of regression coefficients is symbolized by bold beta **β**,

X =

1.000	0.450
1.000	0.000
1.000	0.893
1.000	0.233
1.000	0.000
1.000	0.922
1.000	0.636
1.000	0.304
1.000	0.200
1.000	0.000
1.000	0.749
1.000	0.560

y =

0.496
0.200
0.973
0.519
0.040
0.960
0.700
0.360
-0.130
-0.100
1.150
0.753

β =

β_1
β_2

The matrix algebra expression for the estimate of beta is,

$$\hat{\beta} = [X'X]^{-1} \cdot [X'y]$$

Here it is in the spreadsheet,

XpX = X'X		XpY = X'y	
12	4.947	5.921	
4.947	3.315835	3.909806	
Residual Se:		**0.18694**	← Se
XpX.inverse = $[X'X]^{-1}$		beta	
0.21648	-0.32297	**0.01901**	← $\hat{\beta}_0$ *zero intercept*
-0.32297	**0.78343**	**1.15076**	← $\hat{\beta}_1$ *slope*
	↑$1/SS_x$		

Figure 3. Solving for slope and intercept.

and here are the excel array[1] equations, `

XpX	Xpy
{=MMult(Trans-pose(X.mat),X.mat)}	{=MMult(Trans-pose(X.mat),y)}

XpX.inverse	beta
{=MINVERSE(XpX)}	{=MMULT(XpX.inv,XpY)}

2. Predicted Values and their Standard Errors

The vector of predicted (*smoothed or fitted*) values of y (i.e., points on the trend line) is,

$$\hat{y} = X\hat{\beta}$$

The symbols used above are,

n = the number of data points = the number of rows of X
df.mod = degrees of the model = the number of betas = the number of columns of X
dfe = degrees of freedom for error = n – df.mod = 12-2=10
y_i = measured leak rate for the i[th] leak test run
$\hat{y}_i$ = predicted leak rate for the i[th] leak test run

The standard error of the i[th] predicted value is,

$$S(\hat{y}_i) = Se \cdot \sqrt{1 + x_i[X'X]^{-1}x_i'},$$

where, the symbol x_i is the i[th] row vector of the design matrix **X** and the residual standard error is,

$$Se = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \Big/ dfe} \,.$$

Figure 4. Residual standard error.

In the workbook the vector of prediction standard errors is symbolized *SePred*. Standard errors and their underlying equations are,

SePred	Excel Array[2] Equations
0.272	{=Se*SQRT(1+MMULT(MMULT(INDEX(Xmat,**1**,),XpX.inv),TRANSPOSE(INDEX(Xmat,**1**,)))))}
⋮	⋮
0.285	{=Se*SQRT(1+MMULT(MMULT(INDEX(Xmat,**12**,),XpX.inv),TRANSPOSE(INDEX(Xmat,**12**,)))))}

Finally, the upper and lower prediction limits, which determine the critical value, LC, and the detection limit, LD, are,

$$Prediction\ Limits\colon\ \hat{y} \pm t_{95} \cdot SePred \,.$$

In the workbook the prediction limits and their formulas look like this,

Pred.05	Pred.95	Pred.05	Pred.95
0.184	0.890	{=Yhat - t.95*SePred}	{=Yhat + t.95*SePred}
-0.355	0.393		
0.666	1.428		
-0.070	0.644		
-0.355	0.393		
0.696	1.464		
0.392	1.110	(range[3] of equation)	(range of equation)
0.015	0.723		
-0.109	0.608		
-0.355	0.393		
0.514	1.248		
0.308	1.019		

[2] As written, the top equation does not increment automatically if dragged down; however it will increment if the red 1 is replaced by INDEX(Xmat,ROW()-Xtop,) where the named variable Xtop is the absolute row number of the top row of X.mat.

[3] Select the range, type the equation, without the braces, and press CTRL-Shift-Enter.

3. Decision and Detection Limits

Decision and detection limits are computed on the sheet tabbed "Graph Calculations." The method is to compute Pred.05 and Pred.95 for hypothetical Induced Leak Rates (x-values) at 50 equal steps between zero and the maximum induced leak rate, in this case, 1.90:

True Leak Rate	LR.hat	Se.LR.hat	pred.05	pred.95
0.00000	0.01901	0.20618	-0.35469	0.39272
0.01844	0.04023	0.20520	-0.33167	0.41214
0.03688	0.06145	0.20425	-0.30874	0.43164
0.05532	0.08267	0.20334	-0.28587	0.45122
0.07376	0.10389	0.20247	-0.26308	0.47087
0.09220	0.12511	0.20165	-0.24037	0.49060
0.11064	0.14633	0.20087	-0.21774	0.51041
0.12908	0.16756	0.20014	-0.19518	0.53029
0.14752	0.18878	0.19944	-0.17271	0.55026
0.16596	0.21000	0.19880	-0.15031	0.57031
0.18440	0.23122	0.19819	-0.12800	0.59043
0.20284	0.25244	0.19764	-0.10577	0.61064
⋮	⋮	⋮	⋮	⋮
0.62696	0.74050	0.19779	0.38201	1.09899
0.64540	0.76172	0.19836	0.40219	1.12124
⋮	⋮	⋮	⋮	⋮
0.92200	1.08002	0.21207	0.69565	1.46439

Compute LD and LC		
index	ILR	pred.05
35	0.62696	0.38201
36	0.64540	0.40219
LD	0.63674	
LC	0.39272	

Arrow from LC to LD		
LC	0	0.39272
	0.63674	0.39272
LD	0.63674	0.05
	0.63674	0

The critical value, LC, for raising a leak warning is the value of Pred.95 at true leak rate = 0, in this case LC = 0.50914. The minimum detectable leak rate, LD, is the true leak rate at which Pred.05 = LC. In the spreadsheet it is calculated by linear interpolation. In this case pred.05 is equal to LC somewhere between true leak rates 0.98904 and 1.02708. The spreadsheet approximates LD by linear interpolation:

$$LD \cong \mathbf{0.62696} \cdot \left(\frac{0.40219 - 0.39271}{0.40219 - 0.38201}\right) + \mathbf{0.64540} \cdot \left(\frac{0.39271 - 0.38201}{0.40219 - 0.38201}\right) \cong \mathbf{0.63674}$$

For the record, the exact value is LD = 0.62673; the user can check the accuracy of LD at the bottom of the "Graph Calculations" sheet:

Trial LD ↓	LR.hat	Se.LR.hat	↓ pred.05 - LC
0.63673	0.75174	0.19809	0.00000
↑ Change this value until this is zero ↑			

Algebraic expressions for the critical value L_C and the minimum detectable leak rate L_D are,

$$L_C = \hat{\beta}_0 + t_{df_e}^{95} \cdot S_e \cdot \sqrt{(n+1)/n + \bar{x}^2/SS_x}$$

$$L_D = 2 \cdot \frac{\hat{\beta}_1 \cdot (L_C - \hat{\beta}_0) - \left(t_{df_e}^{95} \cdot S_e\right)^2 \cdot \bar{x}/SS_x}{\hat{\beta}_1{}^2 - \left(t_{df_e}^{95} \cdot S_e\right)^2/SS_x}.$$

Figure 5. Decision & detection limits.

Numerical values of symbols are in Figure 1 & Figure 3; symbol $t_{df_e}^{95}$ is the 95th percentile of Student's t distribution with degrees of freedom $df_e = n - 2$. Inserting those values in the equations of Figure 5 produces the values of L_C and L_D in Figure 6

$$\begin{array}{rclrcl} \hat{\beta}_0 & = & 0.01901 & \bar{x} & = & 0.41225 \\ t_{df_e}^{95} & = & 1.81246 & 1/SS_x & = & 0.78343 \\ S_e & = & 0.18694 & \hat{\beta}_1 & = & 1.15076 \\ n & = & 12 & & & \end{array}$$

$$\begin{aligned} L_C &= 0.01901 + 1.81246 \cdot 0.18694 \cdot \sqrt{13/12 + 0.41225^2 \cdot 0.78343} \\ &= 0.39272 \end{aligned}$$

$$\begin{aligned} L_D &= 2 \cdot \frac{1.15076 \cdot (0.39272 - 0.01901) - (1.81246 \cdot 0.18694)^2 \cdot 0.41225 \cdot 0.78343}{(1.15076)^2 - (1.81246 \cdot 0.18694)^2 \cdot 0.78343} \\ &= 0.63673 \end{aligned}$$

Figure 6. Numerical example for data of Figure 1.

We went through this exercise, not as a practical method of computing LC and LD - that is done automatically by the calibration worksheet - but because the equations in Figure 5 are needed to construct scaling equations in Annex C.

Annex C: Scaling Analysis

"Scaling" means applying the results of the certification of a leak detection system in a particular tank (the certification tank), to a different duration, for example 20 rather than 24 hours, and/or to use in a tank (the candidate tank) with different diameter. This section suggests ways that real-time temperature-corrected changes in product height can be used to develop more efficient ways to estimate leak rate. It also shows how real-time data can be used to develop scaling equations.

1. Scaling Decision and Detection Limits

Scaling involves modifying the critical value, L_C, and the minimum detectable true leak rate, L_D, and possibly changing the duration of a test so that the leak detection system will have 5% probability of a false alarm and at least 95% probability of detecting a leak larger than the (rescaled) minimum detectable true leak rate. To date the commonly used scaling equation, keeping test duration constant, is

$$LD_{candidate} = LD_{certification} \cdot \left(\frac{candidate\ tank\ Surface\ Area}{certification\ tank\ Surface\ Area}\right)$$
$$LC_{candidate} = LD_{candidate}/2$$

To validate or update this equation, we first need to understand how the standard error of a measured leak rate expressed in gallons per hour is influenced by the duration of the test run and the product surface area of the tank.

2. Modeling Sensor Noise

To understand how standard error of a leak rate is influenced by test duration and tank diameter, we need access to real-time temperature-corrected data from sensors during a test run. Figure 7 is the only published[1] example we have found.

Product level change (inches) relative to a reference level is shown on the vertical axis vs. time in hours on the horizontal axis. In the graph, the level appeared to change by about 25 mils in about 27.5 hours; a measured product-level reduction rate (PLR) of about 0.909 mils per hour. According to the certification report[2], the height to volume conversion factor for this run was 3.55 gallons per mil, so the volumetric measured leak rate (LR) is $3.55 \cdot 0.909 \cong 3.23$ gallons per hour.

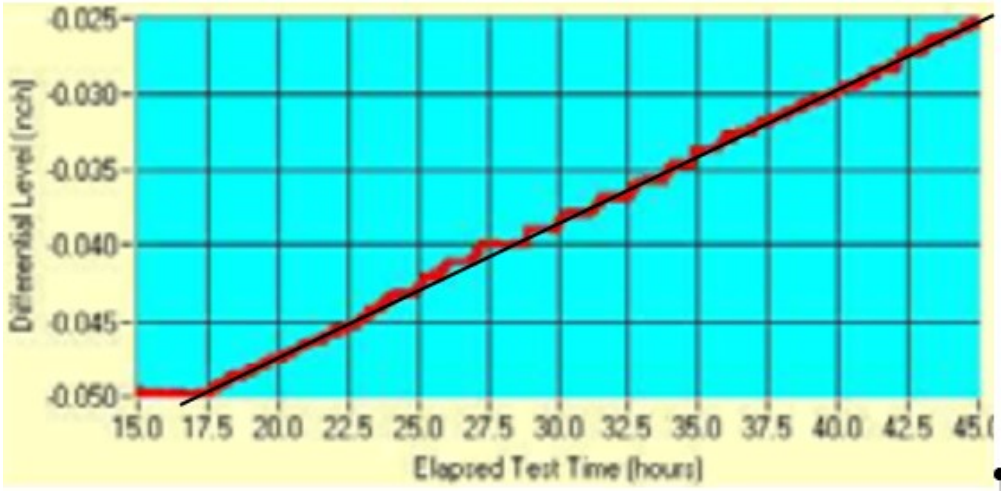

*Figure 7. Real-time output of the LRDP system for Test 1. The time is reported in hours from the beginning of the test (figure 21 of source document**Error! Bookmark not defined.** with added trend line).*

The red trace in in Figure 7 shows long excursions above and below the trend line which are uncharacteristic of white noise. Best statistical practice would therefore be to model the observed noise process to understand how sensor-level noise propagates to leak rate uncertainty and to compute the most efficient (minimum variance) estimate of the leak rate. The ARIMA[3] class of time-series models is the most convenient and most widely used way to do this.

Figure 8, which shows comparable behavior to the trace in Figure 7, was simulated from an ARIMA(0,1,1) model.

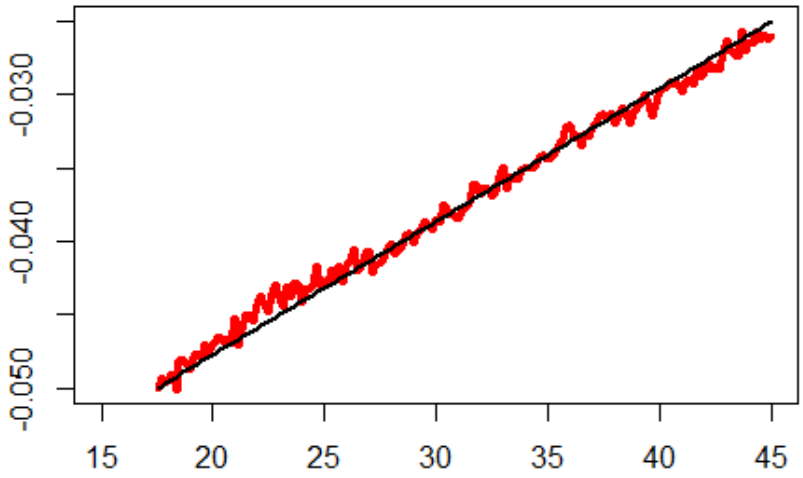

Figure 8 Simulated ARIMA(0,1,1) data for a 0.909 mils per hour leak rate.

The triplet (0,1,1) is called the *order* of the model, generically a model is written ARIMA(p,d,q), where the middle parameter, d, is the number of times the series must be differenced to achieve stationarity (time-homogeneous noise spectrum). Order d = 1 implies that differences of the real-time product level measurements

$$dy_t = (y_t - y_{t-1}) = dt \cdot PLR + stationary\ noise.$$

Symbol dy_t is the incremental change in measured product level between two measurements of the leak rate. PLR is the true product level reduction rate and symbol *dt* is the *sampling interval*, for example 10 minutes between observations in Figure 8. For an ARIMA(0,1,1) model the stationary noise at time t is,

$$e_t = u_t + \theta \cdot u_{t-1},$$

The *innovations* $u_0, \ldots u_n$ are independent and identically distributed random variables with *innovation variance* σ_u^2. Symbol θ is called the *first order moving average coefficient*, or simply the MA1 coefficient. The red trace in Figure 8 was simulated with PLR = 0.909 mils per hour, MA1 coefficient = -0.8[4], slope 0.909 mils per hour and innovation standard deviation $\sigma_u = 0.6\ mils$.

3. Estimating the leak rate

The R program[5] fragment in Figure 9 shows how to estimate the leak rate, moving average parameter θ, and innovation variance σ_u^2 for the simulated data in Figure 8.

```
---------------------------- R Program ----------------------------------
library(forecast)
library(tseries)
y <-
   c(-50.000,-49.302,-49.631,-49.297,-49.066,-50.158,-48.115,-48.049,
                  ⋮       y-coordinates of points in Figure 8      ⋮
     -26.557,-25.596,-26.311,-25.819,-26.220,-25.966)
x <-
   c( 17.500, 17.667, 17.833, 18.000, 18.167, 18.333, 18.500, 18.667,
                  ⋮       x-coordinates of points in Figure 8      ⋮
      43.833, 44.000, 44.167, 44.333, 44.500, 44.667, 44.833, 45.000)

auto.arima(y, d=1, max.p=1,max.q=1,stepwise = TRUE,
             approximation=FALSE,allowdrift = FALSE, allowmean = FALSE,
             trace=TRUE,xreg=x)
---------------------------Statistical Output -------------------------------
Regression with ARIMA(0,1,1) errors

Coefficients:
          ma1    xreg
MA1-> -0.8228  0.8726  <- product level reduction rate (PLR)
s.e.   0.0448  0.0476  <- standard error of PLR (SePLR)

sigma^2 estimated as 0.3158  <- innovation variance
```

Figure 9. R program fragment for estimating loss rate with an ARIMA error model

In the *Statistical Output* section of Figure 9 we find estimates of the MA1 parameter, $\hat{\theta} = -0.82$, product level reduction rate. $\widehat{PLR} = 0.87$, and innovation variance, $\hat{\sigma}_u^2 = 0.32$. The standard error of the estimated leak rate is 0.0476.

The "trace = True" option in the R program in Figure 9 requests the output in Figure 10. The R function "auto.arima" searched for the best fitting model and found ARIMA(0,1,1) to be better than ARIMA(1,1,1) by a tiny margin and overwhelmingly better than any other model. It then used the best model to estimate the leak rate and other parameters as shown in Figure 9

Candidate Model	Figure of demerit (big is bad)
ARIMA(0,1,0)	355.865
ARIMA(1,1,0)	326.362
ARIMA(1,1,1)	284.597
ARIMA(0,1,1)	283.332

Figure 10. Hunt for the best ARIMA model.

The conversion of product level reduction rate, PLR, and its standard error sePLR expressed in mils per hour to leak rate, LR, and its standard error seLR expressed in gallons per hour involves product surface area in square feet:

$$\begin{aligned} LR &= PLR \cdot SurfaceArea \cdot \frac{gals\ per\ cu\ ft}{mils\ per\ ft} \\ &= PLR \cdot SurfaceArea \cdot \mathbf{0.00062338} \end{aligned}$$

$$seLR = sePLR \cdot SurfaceArea \cdot \mathbf{0.00062338}$$

Figure 11 Conversion of Product Level reduction to Product Volume loss.

4. Duration Scaling for ARMA(0,1,1)

The program fragment in Figure 12 shows how we verified the square root (0.5 power) scaling equation for ARMA(0,1,1) models. We ran this program for first order moving average parameter θ values between 0 and 0.8 and in every case the power law exponent rounded to 0.5. So, the equation to convert the certified standard error of estimated PLR to a new test duration is.

$$se_{PLR,new} = se_{PLR.cert} \cdot \sqrt{\frac{duration_{cert}}{duration_{new}}}$$

```
------------------------------ R Program ----------------------------------
library(MASS)
theta = -.8
p <- c(6:9)
n.p <- length(p)
output <- matrix(NA,nrow=n.p,ncol=2)
for (i in 1:n.p) {
  i = 2
  n = 2^p[i]
  C.ma <-  diag(1,n,n); C.ma[row(C.ma) == col(C.ma) - 1] <- rep(theta,n-1)
  Sigma <- t(C.ma) %*% C.ma
  Prec <- ginv(Sigma)
  Se.beta <- 1/sqrt(sum(Prec))
  output[i,] <- c(n,Se.beta)
  weights <- Prec %*% rep(1,n);  weights <- weights/sum(weights)
  Increments <- c(1:n)
  plot(weights ~ Increments, type="p")
}
dur.power <- lm(log(output[,2])~log(output[,1]))
dur.power <- -round(dur.power[[1]][2],1)
paste("Duration Scaling: ",
      "LD.new = LD.cert x (Duration.cert/Duration.new)^",
       dur.power,sep="")
------------------------------ Output -------------------------------------
"Duration Scaling: LD.new = LD.cert x (Duration.cert/Duration.new)^0.5"
```

Figure 12. Calculate Duration Scaling Equation for ARMA(0,1,1) with $\theta = -.8$

5. Surface Area Scaling

The reasoning here is simpler, leak rate (LR) in gallons per hour is product level reduction rate (PLR) in mils per hour times a conversion factor that depends only on the surface area of the tank (see Figure 13. It seems likely that measurement errors at the sensor level depend only on product level, which hardly changes during a 24-hour test and not on product level reduction rate (PLR); therefore, a model for pressure or product-level changes in a different-diameter "target" tank is,

$$dy_t = (y_t - y_{t-1}) = dt \cdot PLR_{target} + stationary\ noise.$$

The $stationary\ noise$ model is the same in the calibration and target tanks: same duration, same interval between observations, same parameters. The only change is the factor in Figure 11 that

converts PLR and se_{PLR} from mils per hour to $LR\ and\ se_{LR}$ in gallons per hour, which is directly proportional to surface area.

$$\begin{aligned} se_{LR,target} &= se_{PLR} \cdot Area_{target} \cdot 0.00062338 \\ &= \frac{se_{LR,cert}}{Area_{cert} \cdot 0.00062338} \cdot Area_{target} \cdot 0.00062338 \\ &= se_{LR,cert} \cdot \left(\frac{Area_{target}}{Area_{cert}}\right) \end{aligned}$$

Combining the duration analysis in Section 4 of this Annex and the diameter analyses in this Section, we get the scaling equation for leak rate standard error,

$$se_{LR,target} = se_{LR,cert} \cdot \left(\frac{Area_{target}}{Area_{cert}}\right) \cdot \sqrt{\frac{Duration_{cert}}{Duration_{target}}}$$

Figure 13.

6. Implications for Scaling L_C and L_D

Our proposal for scaling LD and LC to tightness tests in a tank other than the calibration tank rests on the assumption that residual standard error (from regression calibration, symbol "Se" in Figure 4) is a good approximation of $se_{LR,cert}$ (from an ARIMA fit to real-time product level measurements as in Figure 9).

Given that assumption, scaling equations for regression calibration statistics are,

$$\begin{aligned} Se_{target} &\cong Se \cdot \left(\frac{Area_{target}}{Area_{cert}}\right) \cdot \sqrt{\frac{Duration_{cert}}{Duration_{target}}} \\ \hat{\beta}_{0,target} &\cong \hat{\beta}_0 \\ \hat{\beta}_{1,target} &\cong \hat{\beta}_1 \end{aligned}$$

Finally, we insert these values in Figure 5Figure 6 to get scaling equations for decision and detection limits,

$$L_{C,target} = \hat{\beta}_0 + t_{df_e}^{.95} \cdot Se_{target} \cdot \sqrt{(n+1)/n + \bar{x}^2/SS_x}$$

$$L_{D,target} = 2 \cdot \frac{\hat{\beta}_1 \cdot (L_C - \hat{\beta}_0) - \left(t_{df_e}^{.95} \cdot Se_{target}\right)^2 \cdot \bar{x}/SS_x}{\hat{\beta}_1{}^2 - \left(t_{df_e}^{.95} \cdot Se_{target}\right)^2/SS_x}.$$

7. Implications for Data Reporting

We recommend that real-time temperature and density-correction reduction in product level data be included in the certification report, preferably in mils, inches, or feet of product. If there are multiple sensors, this may be the average real-time reduction in level at each time stamp. We

recommend that these data be recorded at a fixed time interval. The time interval must be constant for all calibration runs but can be any value between 5 and 12 minutes.

[1] "Pre-Production Initiative: Low Range Differential Pressure (LRDP) Leak Detection System — Red Hill Evaluation," Leslie A. Karr, William R. Major, Joseph W. Maresca, Jr., Gary A. Hayter, Michael J. Talhami, James W. Starr, and Richard W. Wise; https://www.dropbox.com/s/lhdtnjy7kfp9gn2/LRDP%20Red%20Hill%20Results%20Paper.pdf?dl=0

[2] Analysis of all 12 measured leak rates reported in endnote note [1]1 indicated a bias of 0.64 gph, so the measured leak rate was reported as 3.23-0.64 = 2.59 gph.

[3] "Autoregressive Integrated Moving Average", Wikipedia, https://en.wikipedia.org/w/index.php?title=Autoregressive_integrated_moving_average&oldid=813878003

[4] This choice of model parameters is based on our analysis of unpublished real-time sensor data.

[5] R Core Team (2017). "R: A language and environment for statistical computing." R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/ .