BIOE 598 Case Study 3: The Trebuchet Simulator

Presenter: Zong Fan

Email: zongfan2@illinois.edu

TEAM-ID: 2AOTEkjg

1. Research Objective

To adjust a trebuchet simulator with 3 parameters to hit a specified distance:

- Fulcrum height (FH)
- Counterweight mass (CM)
- Sling length (SL)

2. Method

- In this case, instead of modeling distance (D) directly with 3 parameters, we could model the relation between the discrepancy with the parameters as:

$$Err = |D_{real} - D_{target}| = f(FH, CM, SL)$$

- Ideally, Err has the minimum value 0, so that the model should fit the curvature.
- Response surfrace methodology is used to fit the nonlinear function f by:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{j=1}^{k} \sum_{i=1}^{j-1} \beta_{ij} x_i x_j \quad \text{k=3}$$

2.1 Hoke Designs

Choose Hoke Designs D2 for RSM, because traininng budget (12 runs) is too small for CCD and BBD.

x_1	x_2	x_3
-1	-1	-1
1	1	-1
1	-1	1
-1	1	1
1	-1	-1
-1	1	-1
-1	-1	1
-1	0	0
0	-1	0
0	0	-1

+

x_1	x_2	x_3
0	0	0
0	0	0

Add another 2 runs at center points to test the lack of fit

2.2 Data Coding

Use coding transformation of data to make all coded variables vary over the same range

Transformation formula: $$X = center(X) + \frac{range(X)}{range(code)}[code]$$

Define the range of the FH, CM, SL to [0.3, 0.6], [8.5, 16.5], [0.3, 0.5], respectively, corresponding to code range [-1, 1].

Code-to-Value:

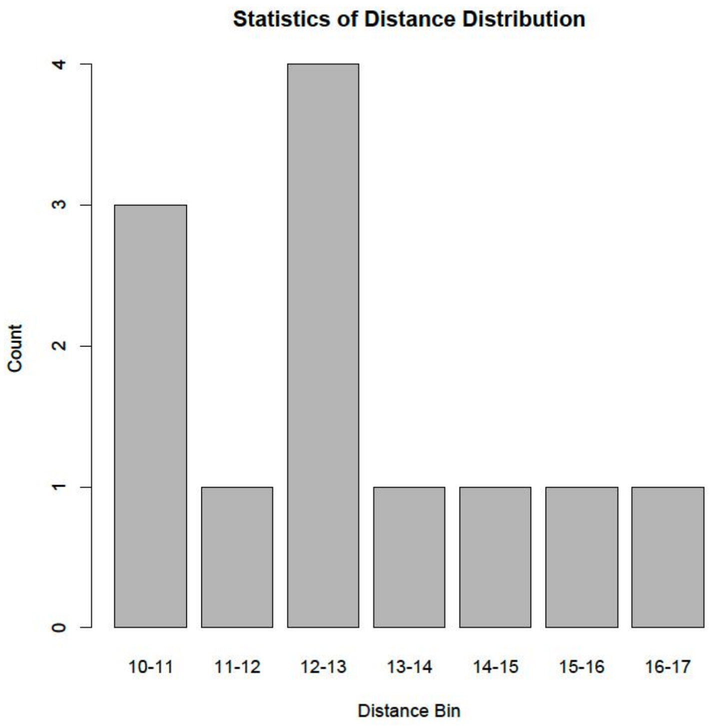
$$FH = 0.45 + 0.15x_1$$
$$CM = 12.5 + 4x_2$$
$$SL = 0.4 + 0.1x_3$$

Value-to-Code:

$$x_1 = (FH - 0.45)/0.15$$
$$x_2 = (CM - 12.5)/4$$
$$x_3 = (SL - 0.4)/0.1$$

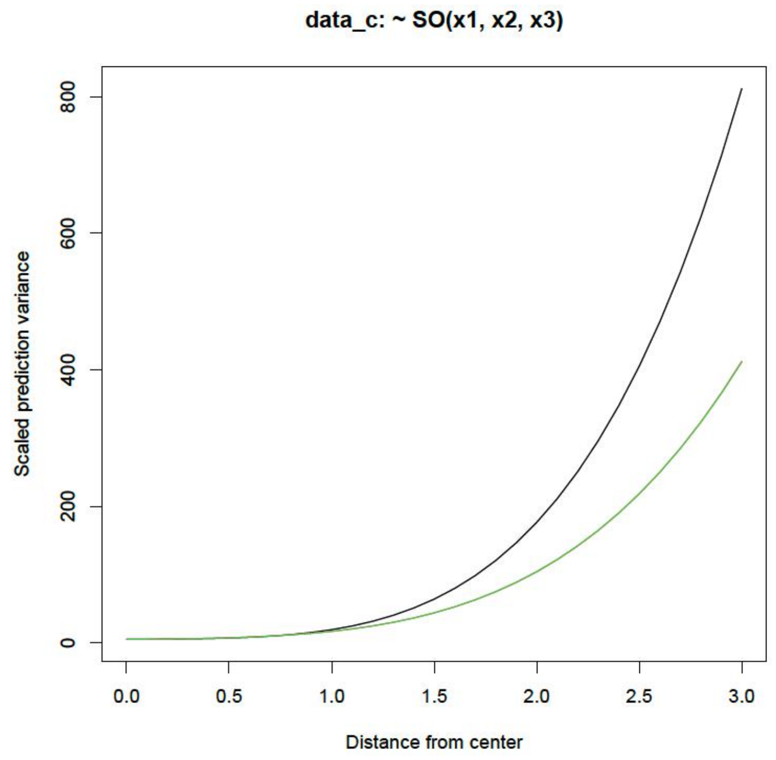
3. Training Data

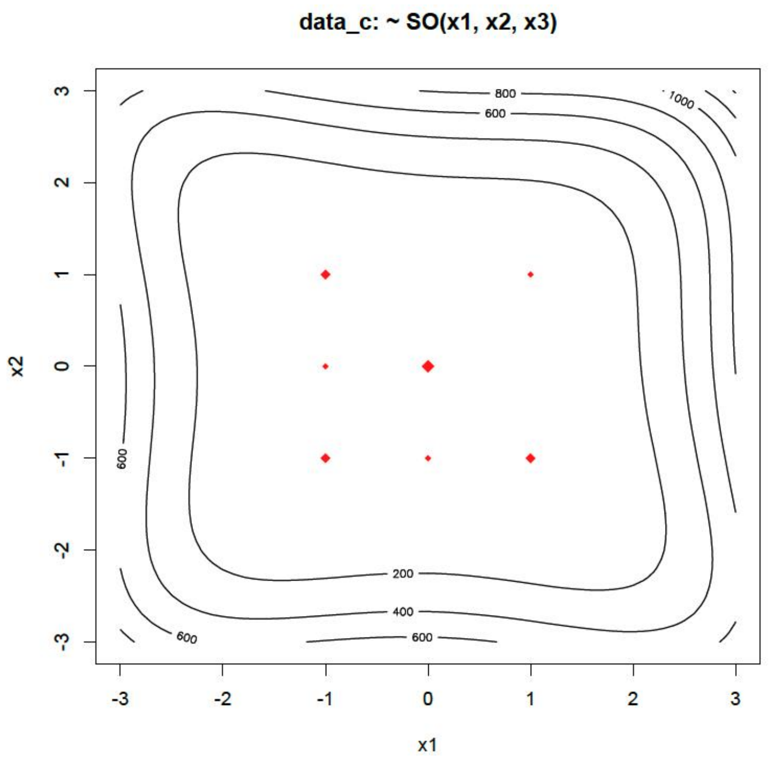
FH	CM	SL	x1	x2	x3	Block	D	ERR
0.3	8.5	0.3	-1	-1	-1	1	10.93	4.07
0.6	16.5	0.3	1	1	-1	1	14.75	0.25
0.6	8.5	0.5	1	-1	1	1	10.4	4.6
0.3	16.5	0.5	-1	1	1	1	12.48	2.52
0.6	8.5	0.3	1	-1	-1	1	11.85	3.15
0.3	16.5	0.3	-1	1	-1	1	12.69	2.31
0.3	8.5	0.5	-1	-1	1	1	13.09	1.91
0.3	12.5	0.4	-1	0	0	1	15.66	0.66
0.45	8.5	0.4	0	-1	0	1	13.01	1.99
0.45	12.5	0.3	0	0	-1	1	13.85	1.15
0.45	12.5	0.4	0	0	0	1	14.84	0.16
0.45	12.5	0.4	0	0	0	1	13.52	1.48



4. Results

4.1 Analysis of uniform precision and rotatable designs





In this case, hoke design has uniform precision but is not rotatable.

4.2 Fitting response-surface model

$$y = 0.7 + 0.2x_1 - 0.53x_2 + 0.46x_3 - 0.26x_1x_2 + 0.93x_1x_3 + 0.62x_2x_3 + 0.27x_1^2 + 0.87x_2^2 + 1.02x_3^2$$

- R-square=0.762, p-value=0.181

- Stationary point is: x1=-3.86, x2=-0.91, x3 =1.82 or FH=-0.13, CM=8.87, SL=0.58

Obviously, the equation is not statistically signifcant, and the FH value at stationary point is < 0, which is also wierd.

4.3 Re-fitting

Rerun RSM after removing statistically and practically insignificant terms:

- TWI(x1, x2)
- PQ(x1)

$$y = 0.79 + 0.28x_1 - 0.39x_2 + 0.59x_3 + 1.04x_1x_3 + 0.74x_2x_3 + 1.02x_2^2 + 1.17x_3^2$$

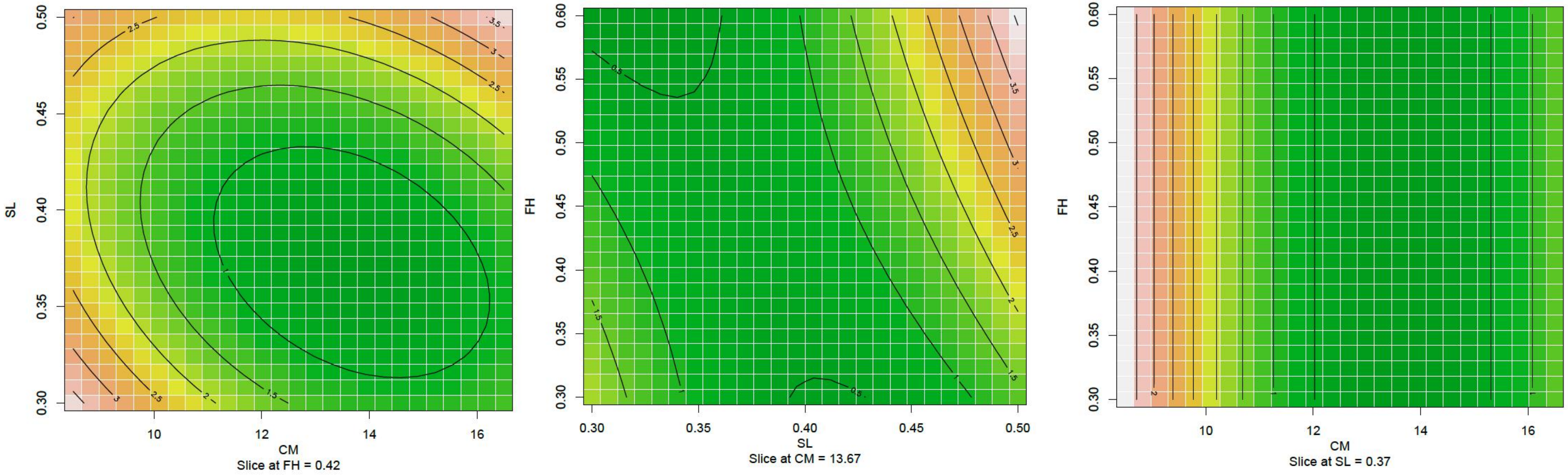
R-square=0.832, p-value=0.026 (Significant!)

Stationary point is: x1=-0.16, x2=0.29, x3 =-0.27 or FH=0.43, CM=13.67, SL=0.37

Eigenvalues: λ1=1.58, λ2=0.82, λ3=-0.22 => Its stationary point is a saddle point.

4.4 Displaying response surface

Fitted response-surface contour plot near the stationary point



Canoical path analysis of the stationary point indicates it seems to be the minimum value. So our strategy to approach Err=0 is jittering around the stationary point. Particularly, freeze SL because its effect size is largest.

4.5 Testing Data

FH	CM	SL	x1	x2	x3	D	ERR
0.42	13.7	0.37	-0.2	0.3	-0.3	15.7	0.7
0.42	13.68	0.37	-0.2	0.3	-0.3	14.04	0.96
0.42	13.69	0.37	-0.2	0.3	-0.3	15.97	0.97
0.42	13.7	0.37	-0.2	0.3	-0.3	15.2	0.2
0.42	13.7	0.37	-0.2	0.3	-0.3	16.06	1.06
0.41	13.7	0.37	-0.27	0.3	-0.3	15.53	0.53
0.41	13.7	0.37	-0.27	0.3	-0.3	14.64	0.36
0.41	13.7	0.37	-0.27	0.3	-0.3	13.7	1.3

mean(ERR) = 0.76, sd(ERR)=0.38

Close to predicted error at the stationary point (ERR_predict=0.627)

5. Conclusion

- With 12 training runs, the estimated response-surface function could predict distance error by 0.776 ± 0.38, close to the value at the stationary point.

- The statistical significance of fitting function could be improved by removing insignificant terms in the second-order function.

- There are actual values > 15, which means the estimated stationary point may not be the minimum. More experiment runs are needed to find the optima.

Appendix: 1. running code

```
library("rsm")
library("daewr")

data <-
read.csv("/Users/zongfan/Downloads/test_data_hoke_4.csv")
data_c <- coded.data(data, x1~(FH-0.45)/0.15,x2~(CM-
12.5)/4,x3~(SL-0.4)/0.1)
SO_model <- rsm(ERR~SO(x1,x2,x3),data=data_c)
SO_model_sim <-
rsm(ERR~FO(x1,x2,x3)+TWI(x1,x3),TWI(x2,x3)+PQ(x2,x3),data
=data_c)
# variation function plot
varfcn(data_c, ~SO(x1,x2,x3), dist=seq(0,3,0.1))
varfcn(data_c, ~SO(x1,x2,x3), dist=seq(0,3,0.1),
contour=TRUE)
summary(SO_model)
summary(SO_model_sim)
# surface response contour plot
contour(SO_model_sim, ~x1+x2+x3, image=TRUE)
contour(SO_model_sim,
x2~x3,image=TRUE,at=data.frame(x1=0.1634500))
contour(SO_model_sim,
x1~x3,image=TRUE,at=data.frame(x2=0.2932412))
contour(SO_model_sim,
x1~x2,image=TRUE,at=data.frame(x3=-0.2711756))

# stationary point
sp <- data.frame(x1=c(-0.1634500), x2=c(0.2932412),
x3=c(-0.2711756))
sp_code <- code2val(sp, codings(data_c))
# predict stationay point value
sp_value <- predict(SO_model_sim, sp)
# canonical path
canonical.path(SO_model_sim)

# get test code
inf_fh <- c(0.42, 0.42, 0.42, 0.42, 0.42, 0.41, 0.41, 0.41)
inf_cm <- c(13.7, 13.68, 13.69, 13.7, 13.7, 13.7, 13.7,
13.7)
inf_sl <- c(0.37, 0.37, 0.37, 0.37, 0.37, 0.37, 0.37, 0.37)
inf_data <- data.frame(FH=inf_fh, CM=inf_cm, SL=inf_sl)
inf_code <- val2code(inf_data, codings(data_c))
```

2. RSM model summary

SO(x1,x2,x3)

```
Call:
rsm(formula = ERR ~ SO(x1, x2, x3), data = data_c)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.70333    0.39851  1.7649   0.2196
x1           0.20125    0.32855  0.6125   0.6025
x2          -0.52875    0.32855 -1.6094   0.2488
x3           0.45875    0.32855  1.3963   0.2974
x1:x2       -0.25583    0.33785 -0.7572   0.5280
x1:x3        0.93167    0.33785  2.7576   0.1102
x2:x3        0.62167    0.33785  1.8400   0.2071
x1^2         0.27458    0.62118  0.4420   0.7017
x2^2         0.87458    0.62118  1.4079   0.2945
x3^2         1.02208    0.62118  1.6454   0.2416

Multiple R-squared:  0.9566,  Adjusted R-squared:  0.7615
F-statistic: 4.902 on 9 and 2 DF,  p-value: 0.1809
```

```
Analysis of Variance Table

Response: ERR
                Df Sum Sq Mean Sq F value Pr(>F)
FO(x1, x2, x3)   3 5.8307 1.94357  4.0794 0.2031
TWI(x1, x2, x3)  3 6.1658 2.05528  4.3139 0.1939
PQ(x1, x2, x3)   3 9.0241 3.00803  6.3136 0.1398
Residuals        2 0.9529 0.47643
Lack of fit      1 0.0817 0.08167  0.0937 0.8109
Pure error       1 0.8712 0.87120

Stationary point of response surface:
       x1        x2        x3
-3.864241 -0.907128  1.812654

Stationary point in original units:
        FH         CM         SL
-0.1296362  8.8714879  0.5812654

Eigenanalysis:
eigen() decomposition
$values
[1]  1.36089110  0.83211179 -0.02175289

$vectors
         [,1]       [,2]       [,3]
x1 0.3053040 -0.4428534  0.8430126
x2 0.4545339  0.8456926  0.2796481
x3 0.8367726 -0.2978001 -0.4594852
```

FO(x1,x2,x3)+TWI(x1,x3)+TWI(x2,x3)+PQ(x2,x3)

```
Call:
rsm(formula = ERR ~ FO(x1, x2, x3) + TWI(x1, x3) + TWI(x2, x3) +
    PQ(x2, x3), data = data_c)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.78855    0.32293  2.4419  0.07107 .
x1           0.28444    0.23623  1.2041  0.29494
x2          -0.39872    0.24213 -1.6468  0.17495
x3           0.58878    0.24213  2.4317  0.07185 .
x1:x3        1.04893    0.24859  4.2196  0.01348 *
x2:x3        0.73893    0.24859  2.9725  0.04104 *
x2^2         1.02152    0.47781  2.1379  0.09932 .
x3^2         1.16902    0.47781  2.4466  0.07070 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.939,   Adjusted R-squared:  0.8324
F-statistic: 8.803 on 7 and 4 DF,  p-value: 0.02639
```

```
Analysis of Variance Table

Response: ERR
               Df  Sum Sq Mean Sq F value  Pr(>F)
FO(x1, x2, x3)  3  5.8307  1.9436  5.8045 0.06120
TWI(x1, x3)     1  2.9811  2.9811  8.9030 0.04059
TWI(x2, x3)     1  1.7588  1.7588  5.2528 0.08368
PQ(x2, x3)      2 10.0635  5.0318 15.0274 0.01380
Residuals       4  1.3394  0.3348
Lack of fit     3  0.4682  0.1561  0.1791 0.90085
Pure error      1  0.8712  0.8712

Stationary point of response surface:
        x1         x2         x3
-0.1634500  0.2932412 -0.2711756

Stationary point in original units:
        FH         CM         SL
 0.4254825 13.6729646  0.3728824

Eigenanalysis:
eigen() decomposition
$values
[1]  1.584875  0.821474 -0.215819

$vectors
         [,1]       [,2]       [,3]
x1 -0.2666957  0.2908389  0.9188504
x2 -0.5285449 -0.8413639  0.1129027
x3 -0.8059241  0.4555430 -0.3781095
```