Automated Projectile Design Software

Mark Steinhoff
Arrow Tech Associates
Projectile Design

• Ballistic Projectile Design
  – Performance Specs
  – System Interface
  – Candidate Design
    • Mass Properties
    • Aerodynamics
    • Ballistic Effectiveness
    • Payload Effectiveness
Projectile Design

• Ballistic Projectile Design
  – Performance Specs
  – System Interface
  – Candidate Design
    • Mass Properties
    • Aerodynamics
    • Ballistic Effectiveness
    • Payload Effectiveness

• Guided Projectiles
  – Same as ballistic plus
  – Control mechanisms
  – Sensors
  – Autopilot
  – Guidance strategy
Mission

• Ballistic Mission
  – How often will you hit the target ($P_h$)
  – When you hit it, what is the likelihood of a kill ($P_{k/h}$)
Mission

• Ballistic Mission
  — How often will you hit the target ($P_h$)
  — When you hit it, what is the likelihood of a kill ($P_{k/h}$)

• Guided
  — Same as Ballistic, plus
  — Remove system errors
  — Trajectory shaping
    • Glide for extended range
    • Dive to clear obstacles or for lethality
Bottom Line

• You now have to:
  – Design the projectile
  – Decide on a control mechanism
  – Design the auto pilot
  – Implement a Guidance strategy
Commercial/Military Projectile Design Tools

- Custom/Proprietary Software
  - Developer uses different analysis modules handing off data from one to the other
  - Stand alone modelers
    - Model building (PRO-E or Solid Works)
    - Aerodynamic estimation (CFD, Missile DATCOM, MILS3 or AP)
  - Simulation codes
    - Hand coded custom solutions
    - Typically Project A evolves into Project B evolves into Project C

- PRODAS
  - Legacy codes embedded into an integrated software system
  - Validated simulations
  - Macro language

- MATLAB/Simulink
  - Like Legacy Simulation codes except within an environment
  - Pre-built simulation blocks and integration engines
# Software Metrics

<table>
<thead>
<tr>
<th>Category</th>
<th>Custom</th>
<th>PRODAS</th>
<th>MATLAB/Simulink</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity Of Models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed of Execution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Push to Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed/Cost of Implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-Validated</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
First Level - Conceptual Design Studies (Proposal)
- For a Given Projectile, What Improvements in Performance Can be Obtained IF a Control Force and/or Moment is Available?
- Simple to model and assess the Benefit of a Flight Control System

Second Level - Detailed Design Studies (Design)
- Perform Parametric Trade Studies to design the details of the Control Mechanism
- Assess the Performance of a Smart Weapon

Third Level - Final Detailed Design (Test)
- At this Stage, Detailed Models of the Sensor Suite and Control Law are Included in Analysis
- Models will include real time loop rates
- Model should generate C code for embedded processors
Guided Projectile Development Cycle

Proposal

- Requirements
- Concepts
- Trades
- System Evaluation

PRODAS
Model, predict Aeros, Trajectories and System Effectiveness
Guided Projectile Development Cycle

- Requirements
- Concepts
- Trades
- System Evaluation

**Proposal**

- PRODAS
  Model, predict Aeros, Trajectories and System Effectiveness

**Design**

- System Design
- Component Design
- GN&C Design

Preliminary GN&C Development
PRODAS
Guided Projectile Development Cycle

Proposal

- Requirements
- Concepts
- Trades
- System Evaluation

PRODAS
Model, predict Aeros, Trajectories and System Effectiveness

Design

- System Design
- Component Design
- GN&C Design

Preliminary GN&C Development
PRODAS

Final GN&C Development
MATLAB

MATLAB
Overall System Simulation

ARROW TECH
Guided Projectile Development Cycle

Proposal

Requirements
Concepts
Trades
System Evaluation

PRODAS
Model, predict Aeros, Trajectories and System Effectiveness

Design

System Design
Component Design
GN&C Design

Preliminary GN&C Development
Final GN&C Development
MATLAB
Overall System Simulation

Test

Component Test
Window Tunnel Test
Ballistic Test
Open Loop Test
Closed Loop Test

PRODAS
to predict Tests and evaluate Aero Test Data

MATLAB
HIL and final GN&C Tuning
Example #1 Ballistic Projectile

- Design a 50 caliber projectile that will minimize wind sensitivity at 1000m
  - Start with basic shape
  - Vary boat tail length and Ogive length and shape
Subtle Changes to Ogive Shape

154mm radius

254mm radius

664cm radius
Not So Subtle Changes to Ogive Length

Standard

-3mm

+3mm
Not So Subtle Changes to Boat Tail

- Standard
- +15%
- +30%
Example #1 Metrics

• 27 designs evaluated
• 7 analysis modules executed
• 50 seconds run time
• 200 lines of PRODAS macro code
• 4 hours to develop
• 1 Excel file of results

• Results:
  – Decreased wind sensitivity by 7.6%
  – Some configurations increased by as much as 10%
Example #1 Extended Metrics

- Started with previous macro
- 125 designs evaluated
- 7 analysis modules executed
- 4 minutes run time
- 212 lines of PRODAS macro code
- 2 minutes to modify
- 1 Excel file of results

- Results:
  - Decreased wind sensitivity by 7.8%
  - Some configurations increased by as much as 25%
Example #2 Guided Projectile

• Basic design of mortar body is fixed, evaluate different fin/canard designs to meet multiple requirements.
  – Find maximum ballistic range
  – Find maximum gliding range using open loop control
  – Find maximum target range with vertical impact using open loop control
Analysis Map

Base line Plus
- +/- 10% and 20% Fin span
- +/- 10% and 20% Canard span
- +/- 10% and 20% Canard chord

Vary Fin and Canard Dimensions

Estimate Aerodynamics

125 Different Projectile Designs

New Projectile
New Projectile
New Projectile
New Projectile Data File

Data Collection

Maximum Ballistic Range
Maximum Gliding Range
Maximum Vertical Impact

ARROW TECH
Macro Map

Macro #1
Base line Plus
- +/- 10% and 20% Fin span
- +/- 10% and 20% Canard span
- +/- 10% and 20% Canard chord

Vary Fin and Canard Dimensions

Estimate Aerodynamics

Data Collection

125 Different Projectile Designs

New Projectile
New Projectile
New Projectile
New Projectile Data File

Macro #2
Maximum Ballistic Range

Macro #3
Maximum Gliding Range

Macro #4
Maximum Vertical Impact
Guided Flight Macros

- Simple open loop canard controller embedded in the GN&C Prototype tool
- Maximum Gliding Range Macro
  - Macro to iterate these design variables:
    - Quadrant elevation
    - Time Glide on
    - Canard application level to limit total AOA
- Maximum Range with Vertical Impact
  - Macro to iterate these design variables:
    - Quadrant elevation
    - Time Glide on
    - Time Dive on
    - Canard application level to limit total AOA
Example #2 Results

• About half of the configurations unstable
• 40% met the ballistic requirement
• 15% met the extended range
• 8% met all the requirements

• Iterated this analysis with three different air frames
Conclusions

• Thorough ballistic development is tough
  – Automation lessens the burden
  – Guided projectiles are even worse

• Match the tool to the job
  – Where are you in the development cycle?
  – Fast or Detailed?
  – Do you have to validate the sims?

• Tools are readily available
Example #1 PRODAS Script

• The following script is included as an example showing how to modify your model with code.

• It will:
  – Modify a base model
  – Calculate Mass Properties
  – Estimate muzzle velocity using Frankfort Interior ballistics
  – Estimate aerodynamics using Spinner 2000
  – Fly a 6DOF trajectory with no wind
  – Estimate stability at cold temperatures
  – Fly another trajectory with 10 knot cross wind
  – Prepare of comma separated value file with the results

• If you have questions or comments, please contact
  – Mark Steinhoff, Arrow Tech Associates (802) 865-3460 ext 18
How to Run this Script

• Open a new PRODAS macro window
• Copy and Paste the lines from the following slides
• Change the path and projectile name of interest
  – The projectile model should include a case and propellant
• The script expects certain elements to be named in your projectile model
  – Name the:
    • Ogive outer skin – OG
    • Ogive void element – OGV
    • Lead that fills the Ogive – OGL
    • Body outer skin – BD
    • Body void element – BDV
    • Lead that fills the Body – BDL
    • Boat tail outer skin – BT
    • Boat tail void element – BTV
    • Lead that fills the Boat tail – BTL
    • Void in the propellant for the Boat tail - PROPV
SUB MAIN

"PRODAS MACRO SCRIPT FILE 12/14/09
FILENAME= T:\PRODASV35\SCRIPTS\BUILDERV1.PVB"

PROJDIR = "R:\ARROW TECH AUTHORED PAPERS 2010\"
TESTPROJ = PROJDIR & "50 CAL BALLISTIC SNIPER PROJECT.PR3"

'INITIALIZE AN ARRAY TO STORE THE SUMMARY TABLE
DIM LINEHDR(50)
DIM LINECOL(50)
ACTIVECOL=20

LINEHDR(0)="CONFIG"
LINEHDR(1)="MASS"
LINEHDR(2)="IX"
LINEHDR(3)="IY"
LINEHDR(4)="CG"
LINEHDR(5)="PROP"
LINEHDR(6)="CHAMBER"
LINEHDR(7)="MV"
LINEHDR(8)="X1"
LINEHDR(9)="Y1"
LINEHDR(10)="Z1"
LINEHDR(11)="MACH"
LINEHDR(12)="GYRO"
LINEHDR(13)="X2"
LINEHDR(14)="Y2"
LINEHDR(15)="Z2"
LINEHDR(16)="TOF"
LINEHDR(17)="X ERROR"
LINEHDR(18)="Y ERROR"
LINEHDR(19)="Z ERROR"
LINEHDR(20)="RMS ERROR"
'INITIALIZE THE TEXT FILE TO ACCUMULATE THE RESULTS
SET RESULTS = MACROSYSTEM.INITIALIZERESULTSFILE
RESULTSFILENAME = PROJDIR & "RESULT " & MONTH(DATE) & ", " & DAY(DATE) & " " & HOUR(TIME) & ", " & MINUTE(TIME) & ".TXT"

RESULTS.OPENFILE RESULTSFILENAME
'Reset the RESULTS PATH
RESULTS.WRITEHEADER
LINEOUT= ""
RESULTS.WRITESTRING LINEOUT
FOR J = 0 TO ACTIVECOL
   LINEOUT= LINEOUT & LINEHDR(J)
   IF J < ACTIVECOL THEN LINEOUT=LINEOUT & CHR(9)
NEXT
RESULTS.WRITESTRING LINEOUT
FOR BTINDEX= 1 TO 3
   FOR OGLINDEX = 1 TO 3
      FOR OGRINDEX = 1 TO 3
         SET PROJ = MACROSYSTEM.INITIALIZEPROJECTILE
         IOPEN= PROJ.OPENDATAFILE(TESTPROJ)
         PROJ.FORCEUNLOCKPROJECTILE 'MAKE SURE IT IS READY TO BE CHANGED
         SET MODEL = PROJ.MODEL("SYSTEM")
         SET BT = MODEL.RETURNNAMEDELEMENT("BT")
         SELECT CASE BTINDEX
            CASE 1 'DO NOTHING
               DELTALENGTH = 0.0
            CASE 2
               DELTALENGTH = BT.LENGTH * 0.15
            CASE 3
               DELTALENGTH = BT.LENGTH * 0.3
         END SELECT
         BTTAG="BT+" & DELTALENGTH*1000 & "MM|
Example PRODAS Script
SET BTV = MODEL.RETURNNAMEDELEMENT("BTV")
SET BTL = MODEL.RETURNNAMEDELEMENT("BTL")
SET PROPV = MODEL.RETURNNAMEDELEMENT("PROPV")
BT.LENGTH = BT.LENGTH + DELTALENGTH
BT.REF_LENGTH= BT.REF_LENGTH- DELTALENGTH
BTV.LENGTH = BTV.LENGTH + DELTALENGTH
BTV.REF_LENGTH= BTV.REF_LENGTH- DELTALENGTH
BTL.LENGTH = BTL.LENGTH + DELTALENGTH
BTL.REF_LENGTH= BTL.REF_LENGTH- DELTALENGTH
PROPV.LENGTH = PROPV.LENGTH + DELTALENGTH
PROPV.REF_LENGTH= PROPV.REF_LENGTH- DELTALENGTH

SELECT CASE OGLINDEX
    CASE 1   'DO NOTHING
        DELTALENGTH = 0.00   'M
    CASE 2
        DELTALENGTH =-0.003  'M
    CASE 3
        DELTALENGTH = 0.003  'M
END SELECT
OGLTAG="OG +" & DELTALENGTH*1000 & "MM|"
SET OG = MODEL.RETURNNAMEDELEMENT("OG")
SET OGV = MODEL.RETURNNAMEDELEMENT("OGV")
SET OGL = MODEL.RETURNNAMEDELEMENT("OGL")
SET BD = MODEL.RETURNNAMEDELEMENT("BD")
SET BDV = MODEL.RETURNNAMEDELEMENT("BDV")
SET BDL = MODEL.RETURNNAMEDELEMENT("BDL")
OG.LENGTH = OG.LENGTH + DELTALENGTH
OG.REF_LENGTH= OG.REF_LENGTH - DELTALENGTH
OGL.LENGTH = OGL.LENGTH + DELTALENGTH
OGL.REF_LENGTH= OGL.REF_LENGTH - DELTALENGTH
BD.LENGTH = BD.LENGTH - DELTALENGTH
BDV.LENGTH = BDV.LENGTH - DELTALENGTH
BDL.LENGTH = BDL.LENGTH - DELTALENGTH
SELECT CASE OGRINDEX
    CASE 1
        'DO NOTHING
        DELTARADIUS= 0.0
        ' MUST BE IN METERS
    CASE 2
        DELTARADIUS= -100.0/1000.0
        ' MUST BE IN METERS
    CASE 3
        DELTARADIUS= 400.0/1000.0
        ' MUST BE IN METERS
END SELECT
SET OG = MODEL.RETURNNAMEDELEMENT("OG")
OG.RADIUS = OG.RADIUS + DELTARADIUS
' MUST BE IN METERS
OGRTAG="OGR + & DELTARADIUS *1000 & "MM|"

INITIALPROP_VOLUME= PROJ.GETDATAPointVALUE("MASSPROP","CALC_PROPELLANT_VOLUME")

PROJ.EXECUTEANALYSIS "MASS2000"
WEIGHT= PROJ.GETDATAPointVALUE("MASSPROP","CALC_FLY_WEIGHT")
AXIALINERTIA= PROJ.GETDATAPointVALUE("MASSPROP","CALC_FLY_AXIALINERTIA")
TRANSINERTIA= PROJ.GETDATAPointVALUE("MASSPROP","CALC_FLY_TRANSINERTIA")
CGNOSE= PROJ.GETDATAPointVALUE("MASSPROP","CALC_FLY_CGNOSE")
LINECOL(0)=BTTAG & OGTAG & OGRTAG
LINECOL(1)=WEIGHT *1000
LINECOL(2)=AXIALINERTIA
LINECOL(3)=TRANSINERTIA
LINECOL(4)=CGNOSE *1000

PROP_WEIGHT= PROJ.GETDATAPointVALUE("MASSPROP","CALC_PROPELLANT_WEIGHT")
PROP_VOLUME= PROJ.GETDATAPointVALUE("MASSPROP","CALC_PROPELLANT_VOLUME")
LINECOL(5)=PROP_WEIGHT *1000
LINECOL(6)=PROP_VOLUME*100*100*100
Example PRODAS Script

Page 5

PROJ.SETDATAPointVALUE "GUNINFO","CHAMBERVOLUME",PROP_VOLUME
PROJ.SETDATAPointVALUE "INTERIORBALLISTICS","PROPMASS(FRANKFORT)",PROP_WEIGHT

PROJ.EXECUTEANALYSIS "IBAL2000FRANKFORT"

MV = PROJ.GETDATAPointVALUE("TRAJECTORY","MUZZLEVELOCITY")
LINECOL(7)=MV

PROJ.EXECUTEANALYSIS "SPIN2000"

PROJ.SETDATAPointVALUE "TRAJECTORY","RANGEFINAL",1000
PROJ.SETDATAPointVALUE "MET","METTYPE",3 'SET TO STD
PROJ.EXECUTEANALYSIS "TRAJ20006D"
SET TABLE= PROJ.OPENDATATABLE("TRAJECTORY","TRAJRESULTSDATA") 'INIT A DATA TABLE OBJECT
LAUNCHMACH = TABLE.CELLVALUE(1,11)
X1 = TABLE.CELLVALUE(TABLE.ROWS,2)
Y1 = TABLE.CELLVALUE(TABLE.ROWS,3)
Z1 = TABLE.CELLVALUE(TABLE.ROWS,4)
LINECOL(8)=X1
LINECOL(9)=Y1
LINECOL(10)=Z1

PROJ.SETDATAPointVALUE "MET","METTYPE",1 'SET TO COLD

PROJ.EXECUTEANALYSIS "SPINSTAB2000"
SET TABLE= PROJ.OPENDATATABLE("AEROSTABILITY","STABILITYBASIC") 'INIT A DATA TABLE OBJECT
IF LAUNCHMACH < TABLE.CELLVALUE(1,1) THEN
   MYGYRO = TABLE.CELLVALUE(1,2)
ELSEIF LAUNCHMACH > TABLE.CELLVALUE(30,1) THEN
   MYGYRO = TABLE.CELLVALUE(30,2)
ELSE
   FOR I=2 TO TABLE.ROWS
      IF LAUNCHMACH < TABLE.CELLVALUE(I,1) THEN
         RATIO = (TABLE.CELLVALUE(I,1)-LAUNCHMACH )/(TABLE.CELLVALUE(I,1)-TABLE.CELLVALUE(I-1,1))
         MYGYRO = TABLE.CELLVALUE(I,2) - RATIO * ( TABLE.CELLVALUE(I,2) - TABLE.CELLVALUE(I,1))
         EXIT FOR
      END IF
   NEXT
END IF
Example PRODAS Script

LINECOL(11)=LAUNCHMACH
LINECOL(12)=MYGYRO

PROJ.SETDATAPointVALUE "MET","PRES AT SEA LEVEL", 1013.3
PROJ.SETDATAPointVALUE "MET","TEMP AT SEA LEVEL", 15
PROJ.SETDATAPointVALUE "MET","WIND DIRECTION", 0.0 'FROM THE NORTH
PROJ.SETDATAPointVALUE "MET","WIND SPEED", 5.14 'M/SEC = 10 KNOTS
PROJ.EXECUTEANALYSIS "MET2000"

PROJ.SETDATAPointVALUE "MET","METTYPE", 6 'USER MET
PROJ.EXECUTEANALYSIS "TRAJ20006D"

SET TABLE= PROJ.OPENDATATABLE("TRAJECTORY","TRAJRESULTSDATA") 'INIT A DATA TABLE OBJECT
X2 =  TABLE.CELLVALUE(TABLE.ROWS,2)
Y2 =  TABLE.CELLVALUE(TABLE.ROWS,3)
Z2 =  TABLE.CELLVALUE(TABLE.ROWS,4)

LINECOL(13)=X2
LINECOL(14)=Y2
LINECOL(15)=Z2
LINECOL(16)= TABLE.CELLVALUE(TABLE.ROWS,1)
LINECOL(17)=X1-X2
LINECOL(18)=Y1-Y2
LINECOL(19)=Z1-Z2
LINECOL(20)= SQRT((X1-X2)^2 + (Y1-Y2)^2 + (Z1-Z2)^2 )

LINEOUT= ""
FOR J = 0 TO ACTIVECOL
    LINEOUT= LINEOUT & LINECOL(J)
    IF J < ACTIVECOL THEN LINEOUT=LINEOUT & CHR(9)
NEXT
RESULTS.WRITESTRING LINEOUT
NEWFILENAME = "50 CAL BALLISTIC SNIPER PROJECT" &BTINDEX & OGLINDEX &OGRINDEX  & ".PR3"
PROJ.SAVEASDATAFILE PROJDIR & NEWFILENAME
PROJ.CLOSEDATAFILE 'YOU ARE DONE WITH THIS PROJECTILE

NEXT
NEXT
NEXT
RESULTS.CLOSEFILE 'CLOSE THE RESULTS FILE
SET RESULTS = NOTHING
MSGBOX "RESULTS FILE CAN BE FOUND IN " & RESULTSFILENAME

END SUB