Final Report

On the use of Computational Fluid Dynamics and Kriging for the Trajectory Prediction and Fin Optimisation of a Countermeasure

Benjamin Johnson

2015

3rd Year Individual Project

I certify that all material in this thesis that is not my own work has been identified and that no material has been included for which a degree has previously been conferred on me.

Signed  .....

College of Engineering, Mathematics, and Physical Sciences
University of Exeter

Final Report

ECM3102

Title: On the use of Computational Fluid Dynamics and Kriging for the Trajectory Prediction and Fin Optimisation of a Countermeasure

Word count: 12539

Number of pages: 34

Date of submission: Monday, 27 April 2015

Student Name: Benjamin Johnson

Programme: Meng Mechanical Engineering

Student number: 620006622

Candidate number: 017027

Supervisor: Dr Gavin Tabor

Abstract

This project investigates the method of using Computational Fluid Dynamics (CFD) in predicting the trajectory of a countermeasure and in the optimisation of fins. The aim was to establish the viability of the methods, providing examples of their use and to form a basis for the development of further programs. Chemring Countermeasures provided the geometry and accompanying data. The optimisation of the fins is based on a kriging response surface and Latin-Hypercube sampling, for which a MATLAB program was written. This optimisation used the kriging model to interpolate across the sample space and locate the point with the highest probability of improvement. The kriging model was also used to interpolate values of the drag coefficient and Centre of Pressure of the countermeasure to reduce the number of simulations required during the trajectory prediction. The trajectory prediction used an Adams-Bashforth and Adams-Moulton 5^{th} order predictor-corrector method of time-stepping to find the velocity and displacement of the countermeasure. The use of a drag coefficient, calculated from CFD, acted as a surrogate model for this to reduce the number of simulations. Previous research has looked at predicting trajectories but not in the specific flight conditions faced by the countermeasure. Equally, research has covered the optimisation method but not the exact application of it in a four dimensional optimisation problem.

The outcomes were that the use of CFD alongside time-stepping produced accurate and detailed models of the trajectory in three degrees of freedom (d.o.f). The fin optimisation produced an optimal design which had a 98% certainty that no other design could achieve a 25% improvement on it. The methods themselves were confirmed as being viable and effective for predicting trajectories and optimising fins and a number of programs were produced which enable these methods to be easily used and adjusted for further applications.

Keywords: Kriging, Optimisation, Time-stepping, Trajectory prediction, CFD,

Table of contents

1. Introduction and background

This project, set by Chemring Countermeasures following a summer placement, is to find the behaviour of a naval countermeasure during flight by using Computational Fluid Dynamics (CFD). This will enable greater effectiveness of their design and deployment, thus enabling a greater survival rate of the platform. The aims are to investigate the optimisation of the fin design and to model the flight behaviour of the countermeasure.

The computer model of the countermeasure geometry will be provided by Chemring alongside data such as the thrust profile. This will be analysed using CFD and stepped through time to approximate the speed and displacement, which will be checked against true values. The CFD analysis will provide the drag coefficient for the countermeasure at the different speeds. To save on computational cost a surrogate model will be used alongside this to find the drag force at intermediate values. Any deviation in the drag coefficients between the speeds will be interpolated. Following a basic trajectory prediction with movement in only one degree of freedom (d.o.f), extra analyses will be performed in more degrees of freedom to produce a full model of the trajectory.

The fin optimisation will be achieved by first creating a parameterised design and then following an optimisation algorithm, to achieve a design with low drag and high torque to spin the countermeasure. The fin optimisation algorithm will use a combination of an Orthogonal Latin Hypercube (OLHC) and a kriging response surface to find the predicted values of the optimisation index, as well as the standard error at each predicted point. This enables the calculation of the probability of reaching a target increase in the optimisation index; the location of the highest probability is therefore the next sample point. This method was decided upon following research outlined in the literature review.

The literature review is given in Section 2 and covers topics researched which influence decisions made in this project as well as a review of existing research. Section 3 covers the methodology followed in this project and underlying theory. Section 4 documents the refinement and evolution of the method which occurred during application. Section 5 presents and discusses the results. Section 6 discusses the accuracy of the results and summarises the conclusions drawn from the project alongside making recommendations for further work and applications. Section 7 presents the project management, sustainability considerations and the consideration of risks for the project.

2. Literature review

2.1. Existing Research

The behaviour of countermeasures is modelled by Chemring for developing deployment strategies; this is not done using CFD, instead being based on approximations and empirical data. Studies have been performed on missile shaped projectiles or the optimisation of fins [1-3] but these are generally more focused on stability or supersonic behaviour [3-11].

Despeyroux et al [1] investigated Grid fins, a type of lattice panel used as a control surface in missiles and projectiles, for mach numbers of 2 and 0.9 using CFD. Fresconi et al [2] look at the performance of a novel rotating wing actuator and spin stabilisation. Sahu et al [3] investigated the control performance of a canard-controlled projectile using CFD. These three investigations all focus on the optimisation of fins for projectiles but are for fins used for control purposes and are not on the specific optimisation of the shape; equally, they do not cover the prediction of the trajectory.

Baysal and Eleshaky [4] present a method for aerodynamic optimisation based on a CFD sensitivity analysis algorithm, they apply this to a simplified case to provide an example of its application. The method itself is the main focus of the report, in the same way that the optimisation procedure followed for the fin optimisation section in this report is the main focus of that section – Baysal and Eleshaky make the observation that finding an optimised configuration for the common interest is secondary to the method used to achieve it, due to the application of constraints in the optimisation process. Bhagwandin and Sahu [5] investigated the numerical prediction of the pitch damping stability derivatives of finned projectiles at subsonic and transonic flow regimes. Despirito et al [6] also give a report on the prediction of stability derivatives using CFD. This is again done by Sigal [11] for twelve projectile configurations in a comparison to experimental values.

Hartfield et al [7] discus a method for developing optimised designs of symmetric-centerbody ramjet powered missiles using genetic algorithms (GA), although no discussion of the GA is given. Jameson et al [8] reviewed optimisation techniques based on control theory specifically in compressible flow. Silton [9] uses numerical methods to understand experimental anomalies on a 155mm projectile; this is similar to the idea behind using CFD to model the behaviour of the countermeasure. However, Silton is more specifically engaged with producing numerical results that agree with the experimental ones, rather than an outright prediction of trajectory using the numerical methods. Sturek et al [10] predicted a

complete set of aerodynamic performance parameters to an extensive number of projectile configurations using CFD, including spin and fin stabilised ones. However, this neglects the application to the prediction of the trajectory.

Therefore from these reports it can be suggested that the subsonic analysis of the projectile to predict behaviour and trajectory is novel. Although, it is worth noting that the information on countermeasures and similar projectiles is often confidential and therefore inaccessible.

2.2. Countermeasure Data

The data for the countermeasure has been provided by Chemring, the method they use to predict the flight velocity a 4^{th} order Runge-Kutta time-stepping scheme alongside an approximated drag coefficient of 0.8. This means that the 'true' values are not from observed tests and there is the possibility of the real case deviating from them. This means that the calculation of the trajectory using CFD to find the drag coefficients is novel for this design. Therefore the project holds the possibility of improving the prediction of the flight behaviour. The countermeasure is launched at an angle of 45° and has a flight time of 4.4 seconds. After which a drogue chute is deployed, following the ejection of the rocket motor; this holds the countermeasure tail up. The mass of the countermeasure is 25.48kg with the moment of inertia about the centre of mass of the countermeasure being 6.33kgm^2.

The maximum speed of the projectile is around 180m/s; giving a Reynolds number of $1.55x10^6$ [12] and a mach number of 0.529 [13] – the conditions are turbulent, subsonic flow [14]. The subsonic flow regime means that the flow has a mach number less than 1 almost everywhere. Although some geometries can create local regions exceeding a mach number of 1, introducing compressibility, however the shock regions of supersonic flow are not present. The project may involve finding the centre of pressure; this can be done analytically by the CFD software [15].

2.3. Optimisation

In this instance the output optimisation function is unknown, so analytical methods cannot be used, and the time cost of each sample is high. The most effective way of sampling a space is the Orthogonal Latin Hypercube (OLHC), which enables the most even spread of sample points [16]. To reduce the computational cost of finding the optimum, a surrogate model can be used to approximate the output function [17,18]. One of the most effective surrogate models for an unknown function is the kriging model. Kriging operates by statistical means, where the point of highest probability of improvement is the next sample point [17,19,20]

[21]. This method enables the optimisation to be done in around 11 simulations with an initial sample of 16 simulations [19]; the result is the global optimum and is statistically proven to a given certainty. The most effective method for global optimisation that could be found for cases of high sample cost is that proposed by Jones [19], which consists of a kriging response surface with the evaluation of the probability function to locate the next sample point. Jones presents this in an intuitive manner which can easily be extended into more dimensions, thus this will be the method followed in the optimisation section of this report.

2.4. CFD Techniques and Required Practices

To ensure accurate CFD simulations, the meshing techniques and practices used in industry were researched. The boundary layer mesh is refined using an inflation layer of prism cells [22], specified by the number of cell layers, the growth rate of the layers and the first layer height. The main measure of the boundary layer resolution is the y^+ measurement – the wall unit value at the height of the first layer, see Equation 1 [23]. Refinement of the mesh at the boundary layer is required due to the rapid change in velocity in the region [22].

Equation 1:

$$y^+ = \frac{\rho \cdot U_\tau \cdot \Delta y_1}{\mu} \qquad U_\tau = \sqrt{\frac{\tau_w}{\rho}} \qquad \tau_w = \frac{1}{2} \cdot C_f \cdot \rho \cdot U^2 \qquad C_f = 0.058 \cdot Re^{-0.2}$$

Where:

ρ= Density of the fluid (kgm^{-3})
μ= Dynamic viscosity (m^2s^{-1})
U= Velocity of flow (ms^{-1})
Re= Reynolds number
Δy_1= First cell height (m)
U_τ= Frictional Velocity
τ_w= Wall shear stress
C_f= Skin friction coefficient (empirical)

The exact y^+ value required can vary depending on the turbulence model being used. Some turbulence models use wall functions, such as k-ε, and can reduce the required y^+ value. These can have minimum y^+ values to ensure accurate evaluation of the wall functions [23]. This reduction in required boundary layer resolution can reduce computational cost but can neglect separation, rather than fully resolving the flow [23]. Therefore, the likely flow characteristics and expected level of separation must be established, to consider whether wall functions are appropriate [24].

The y^+ value can only be estimated before the simulation and so the exact value must be checked afterwards [23]. CFD-Post, part of the ANSYS workbench, is capable of calculating the y^+ values, so this was practiced as part of the simulation process. Alongside the y^+ values, the boundary layer should be checked to ensure that it is fully contained within the inflation layer and is not limited by it [25]. This can be done by plotting contours of the eddy viscosity

ratio in the inflation layer; if fully contained the contours should be symmetrical and form a band in the inflation layer [25] – this was also checked as part of the simulations.

Another important consideration is the turbulence model that will be used, for external flows the Spalart-Allmaras (SA) turbulence model and k-ε are commonly used [26]. Both are part of the Reynolds Averaged Navier Stokes (RANS) approach, whereby the velocity is broken down into the mean velocity and fluctuating components [26]. SA is a single-equation model which transports a turbulence parameter created in regions of high vorticity. This means the model is very stable and computationally cheap, but can take a long time to develop the full turbulence cascade [26]. The SA does, however, neglect the use of wall functions and instead fully resolves the flow in the boundary layer [24]. This can increase the computational cost of the mesh but will ensure any boundary layer separation is represented. Conversely, k-ε is a two equation model which evaluates the turbulent kinetic energy (k) and dissipation rate (ε) [26], making it computationally more expensive than SA. However k-ε uses wall functions so can produce a computationally less expensive mesh, but may neglect separation.

Detached Eddy Simulation (DES) and Large Eddy Simulation (LES) are part of the Eddy Simulation turbulence models, which resolve the larger eddies in the flow, and are only used if recirculating regions or a large amount of separation occurs [26]. DES and LES often provide the most accurate turbulence modelling however they are computationally very expensive. In high speed flow the turbulent cascade usually results in any detached eddies being dissipated rapidly reducing their effect on the flow [26]. Thus the models produced by SA and k-ε are often just as accurate in high speeds.

The convergence of the simulations is also an important aspect to consider. For steady state simulations it is necessary to monitor the residuals as well as the output parameters of interest – such as lift or drag [27]. Therefore, the solution is considered to have converged when the residuals level off and are in the order of 10^{-4}-10^{-5}, as well as the output parameter of interest levelling off to within a certain tolerance [27]

3. Methodology and theory

3.1. Time-stepping and Trajectory

3.1.1. Time-stepping Methods

The acceleration of the projectile can be found from balancing the forces acting on it, however the velocity and displacement are of more interest and so the acceleration is expressed as an Ordinary Differential Equation (ODE), as in Equation 2:

$$\boldsymbol{a} = \frac{d\boldsymbol{v}}{dt} = f(v,t) \qquad \text{(Equation 2)}$$

Where: a= Acceleration (ms^2); v= Velocity (ms^{-1}); t= Time (s)

To find the velocity it is then possible to use ODE methods. Explicit methods are computationally cheaper but have conditional stability, so errors inherent in the approximation can grow under certain conditions such as a large time-step. Some functions are unconditionally unstable when solved by explicit schemes. Implicit methods are unconditionally stable; these operate by creating a matrix of the points used and the function evaluation at them, then the matrix is inverted simultaneously solving the equations. In this case the function evaluations are to be found by CFD, meaning that a simultaneous solution is not possible. Predictor-corrector methods use an explicit scheme to predict the velocity at the next point; this is then used to find the function evaluation there followed by an implicit scheme correcting the velocity. This method is inherently stable and fixes the issues that would be faced by the explicit and implicit schemes, so will be used in this case.

The most basic time-stepping scheme is the explicit Forward Euler (FE) which in this case would be corrected by the implicit Backward Euler (BE), as in Equation 3. The FE and BE are only first order accurate and are therefore fairly coarse as an approximation.

FE: $v_{n+1}^{p} = v_n + \delta t \cdot f(v_n, t_n)$ BE: $v_{n+1} = v_n + \delta t \cdot f(v_{n+1}^{p}, t_{n+1})$ (Equation 3)

Where: v_{n+1}^{p}= Predicted velocity at time t_{n+1} (ms^{-1}); v_n= Velocity at time t_n; δt= Time step (s)

An improvement on the FE and BE scheme is the Runge-Kutta 4th Order (RK4) explicit scheme, this uses intermediate function evaluations to improve the accuracy, as shown in Equation 4. The RK4 scheme would then be coupled to an implicit corrector.

$$v_{n+1} = v_n + \frac{\delta t}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \qquad \text{(Equation 4)}$$

$$k_1 = f(v_n, t_n)$$
$$k_2 = f\left(v_n + \frac{\delta t}{2} k_1, t_n + \frac{\delta t}{2}\right)$$
$$k_2 = f\left(v_n + \frac{\delta t}{2} k_2,\ t_n + \frac{\delta t}{2}\right)$$
$$k_2 = f(v_n + \delta t \cdot k_3,\ t_n + \delta t)$$

The downside to the RK4 scheme is the additional function evaluations, which mean that it would require more simulations if not run with a surrogate model. To reduce the need for the extra evaluations it is possible to use the explicit Adams-Bashforth (AB) scheme coupled to an implicit Adams-Moulton (AM) corrector. The AB and AM schemes have the benefit that the schemes increase in accuracy as more points become known. For example, the scheme can be initialised with a FE and then adjusted each time step through AB2 to AB5 with no additional function evaluations required. Equally, each of these schemes can be corrected

with the AM schemes in a growing order of accuracy. The AB scheme is shown in Table 1, the corresponding AM schemes are shown in Table 2, where: $f_n = f(v_n, t_n)$.

Due to the high accuracy and few function evaluations, the AB-AM predictor corrector scheme will be used. Although in this case a surrogate model is being used, so RK4 would be appropriate, further development of this project would be to produce a fully automated CFD model in all degrees of freedom. In such a case, a surrogate model may not be appropriate and so the AB-AM scheme would be required, thus it is favourable to demonstrate its use.

Table 1: Adams-Bashforth schemes and the corresponding order of accuracy [28]

Name	Scheme	Order of Accuracy
FE	$v_{n+1} = v_n + \delta t \cdot f_n$	$O(\delta t)$
AB2	$v_{n+1} = v_n + \frac{\delta t}{2} \cdot (3f_n - f_{n-1})$	$O(\delta t^2)$
AB3	$v_{n+1} = v_n + \frac{\delta t}{12} \cdot (23f_n - 16f_{n-1} + 5f_{n-2})$	$O(\delta t^3)$
AB4	$v_{n+1} = v_n + \frac{\delta t}{24} \cdot (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$	$O(\delta t^4)$
AB5	$v_{n+1} = v_n + \frac{\delta t}{720} \cdot (1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} + 251f_{n-4})$	$O(\delta t^5)$

Table 2: Adams-Moulton schemes and the corresponding order of accuracy [28]

Name	Scheme	Order of Accuracy
BE	$v_{n+1} = v_n + \delta t \cdot f_{n+1}$	$O(\delta t)$
AM2	$v_{n+1} = v_n + \frac{\delta t}{2} \cdot (f_{n+1} - f_n)$	$O(\delta t^2)$
AM3	$v_{n+1} = v_n + \frac{\delta t}{12} \cdot (5f_{n+1} + 8f_n - f_{n-1})$	$O(\delta t^3)$
AM4	$v_{n+1} = v_n + \frac{\delta t}{24} \cdot (9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$	$O(\delta t^4)$
AM5	$v_{n+1} = v_n + \frac{\delta t}{720} \cdot (251f_{n+1} + 646f_n - 264f_{n-1} + 106f_{n-2} - 19f_{n-3})$	$O(\delta t^5)$

3.1.2. Surrogate Modelling

Due to the high computational cost of running a simulation to find the drag force at each velocity it is necessary to use a surrogate model. In this case the surrogate model will be the drag coefficient. Although the drag coefficient is unlikely to change greatly with velocity there may be fluctuations or a rapid decline due to a drag crisis – where the boundary layer becomes turbulent. To account for fluctuations in the drag coefficient, they will be interpolated using a kriging model. This will give the drag coefficient at each velocity, with only having sampled some velocity points.

3.1.3. Projectile Motion

The behaviour of the projectile in flight has been fully characterised in terms of each degree of freedom (d.o.f). Equation 5 shows the resulting motion based on the coordinate system in Figure 1; this can be stepped forward in time using finite differencing to find the behaviour throughout the flight of the projectile. It is worth noting at this point that the projectile used by Chemring is fin-stabilised and not spin-stabilised and therefore the movement in β is negligible.

Equation 5:

$$\begin{pmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & I_\beta & 0 \\ 0 & 0 & 0 & I_\alpha \end{pmatrix} \begin{pmatrix} a_z \\ a_r \\ a_\beta \\ a_\alpha \end{pmatrix} = \begin{pmatrix} F_t \\ 0 \\ F_f \\ 0 \end{pmatrix} - \begin{pmatrix} F_{dz} \\ F_{dr} \\ F_{d\beta} \\ C_p F_{dr} \end{pmatrix} + \begin{pmatrix} -mg\cos\alpha \\ mg\sin\alpha \\ 0 \\ C_g mg\sin\alpha \end{pmatrix}$$

Where: m=mass (kg)
I=moment of inertia (kgm^2)
a = acceleration (ms^{-2})
F_t = thrust force (N)
F_f = fin torque (Nm)
F_d = drag force (N) ($F_{d\beta}$ = torque (Nm))
C_p = distance to centre of pressure from tail (m)
C_g = distance to centre of mass from tail (m)

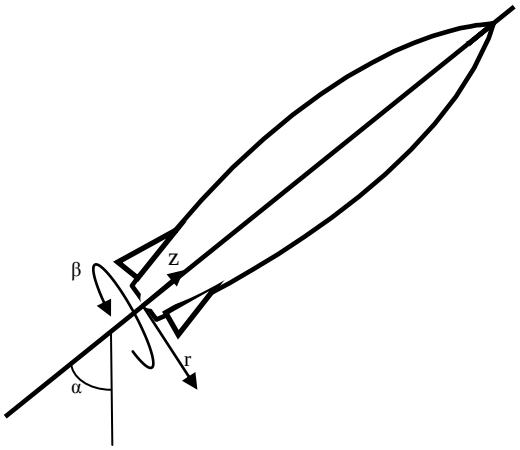


Figure 1: Coordinate system for the degrees of freedom of the projectile flight

It is worth noting that due to rotation in α during the 3 d.o.f case the previous velocities (v_n) must also be adjusted to the new reference frame at each step, such that the adjusted previous velocities, $v_{n_{new}}^{(z)}$ & $v_{n_{new}}^{(r)}$, are given by:

$$v_{n_{new}}^{(z)} = v_n^{(z)}\cos(\Delta\alpha) + v_n^{(r)}\sin(\Delta\alpha)\ ; \qquad v_{n_{new}}^{(r)} = -v_n^{(z)}\sin(\Delta\alpha) + v_n^{(r)}\cos(\Delta\alpha)$$

This adjustment is required due to the reference frame chosen, where Δα is the change in α since the previous time step.

For this project the drag in the z direction was evaluated using CFD at a number of points and then interpolated. However, for the r direction, analytical values for an infinite cylinder were used to demonstrate the method in three d.o.f; meaning that the two dimensions are assumed to be independent. This is not the case since the flow in the r direction could produce separation or increase turbulence, which will increase drag in the z direction. However, this assumption of independence between the two directions should not be too inaccurate, since the majority of the flow velocity is in the z direction, the velocity in the r direction is unlikely to be high enough to cause a large effect on the z direction drag. Also the drag values in the r direction will be inaccurate due to the assumption of it being an infinite cylinder. These two assumptions would be made redundant by performing CFD simulations on the three d.o.f

case. However this would be computationally expensive and so to simply demonstrate the method it is possible to use these assumptions.

3.1.4. Overall Prediction Method

The overall method for the prediction of the projectile motion is to run a simulation at the starting velocity, then use the drag coefficient to predict the forces at future velocities. This gives an idea of the maximum velocity and therefore the velocities which should have simulations run at them to ensure an even distribution of sample points. The drag coefficients will then be found at these points, done in parallel to save on time, and then interpolated using a kriging model. The predicted drag coefficients will then be used at each of their corresponding velocities to find the drag force. The velocity itself will be found using the AB-AM time-stepping method outlined in 3.1.1, as will the displacement. The initial conditions of the trajectory prediction are all zero except: α=45°, v_z=21 ms^{-1}.

3.2. Kriging and Optimisation

3.2.1. Fin Parameterisation

The fin design must be parameterised for the optimisation process. This has been done by taking the fin to be a triangular section of a tube, following the line of the intersection of two cylinders; this gives 4 parameters, shown in Figure 2. The ranges are:

Theta: 30-360°
r_1: 0.005-0.05m,
r_2: 0.05-1m,
A: 5-90°

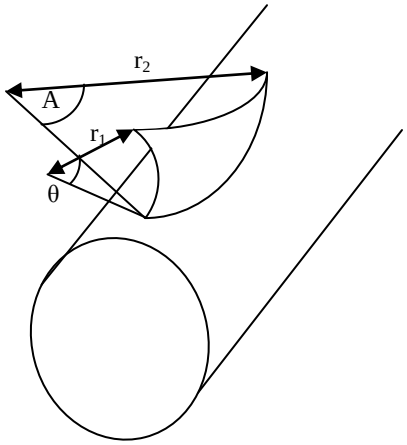


Figure 2: The parameter assignment for the countermeasure fins

3.2.2. Orthogonal Latin Hypercube Construction

The most effective way of sampling a space is the Orthogonal Latin Hypercube (OLHC), this is a specific form of Latin Hypercube (LHC) in which all the points are orthogonal to one another, the standard LHC merely optimises for orthogonality. The construction of an OLHC starts with creating D_k, the $n \times k$ design matrix for the 2^k factorial design [16]. Where k is the number of parameters and is a power of two; n is the number of runs, such that $n = 2^k$. D_k has two levels, 1 and -1, for each factor and therefore contains all possible combinations of 1 and -1 for each parameter [29]. D_k is an orthogonal design, such that $D_K' D_k = nI$, where I is the Identity matrix. A matrix R is a rotation matrix if $R'R = I$, this can be used to create a

rotated version of D_K as $D_R = D_K R$ [16]. It is worth noting that the orthogonality is conserved through rotation since: $D_R' D_R = R' D_K' D_k R = nI$ [16].

To create the rotation matrix R_m, V_m is first constructed whereby [16]:

$$V_0 = [1], \qquad V_m = \begin{bmatrix} V_{m-1} & -(2^m)V_{m-1} \\ (2^m)V_{m-1} & V_{m-1} \end{bmatrix}, \qquad k = 2^m. \qquad \text{(Equation 6)}$$

V_m is then rescaled to make it a rotation matrix, R_m, such that [16]:

$$R_m = \left(\frac{1}{a_m}\right) V_m, \qquad a_m = \left\{\prod_{j=1}^{m} \left(1 + 2^{2j}\right)\right\}^{\frac{1}{2}}. \qquad \text{(Equation 7)}$$

The OLHC, D_R, is then created by [16]: $D_R = D_K R_m$ (Equation 8)

The unit OLHC is formed by dividing by the maximum value of D_R, this is between 1 and -1 so forms the sample points of the parameters by being multiplied by half the range and adding the midpoint for each parameter column of the OLHC.

3.2.3. Kriging

The optimisation of the fins poses a number of difficulties in that the optimisation is in four dimensions with an unknown output function. This means that the optimisation required is global, not local, as the output function cannot be assumed to be unimodal. In global optimisation the method must be able to consider the whole sample space; this makes it far more complex as simply considering the gradient is not sufficient. Many global optimisation techniques therefore use response surfaces to predict the output function to then find a minimum – kriging is one such response surface. Kriging assumes the output at a sample point, x, behaves as a random variable which is normally distributed with mean, μ, and variance, σ^2. Consider two sample points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with outputs $Y(\boldsymbol{x}_i)$ and $Y(\boldsymbol{x}_j)$; it is expected that $Y(\boldsymbol{x}_i)$ and $Y(\boldsymbol{x}_j)$ will be highly correlated if the distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is small. Therefore in kriging the correlation function is as given in Equation 9 [19]:

$$Corr\left[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)\right] = \exp\left(-\sum_{\ell=1}^{d} \vartheta_\ell \left|\boldsymbol{x}_{i\ell} - \boldsymbol{x}_{j\ell}\right|^{p_\ell}\right) \qquad \text{(Equation 9)}$$

Where, ϑ_ℓ and p_ℓ are the kriging parameters. These are later optimised for the known sample outputs to improve the predictions. It is worth noting that ϑ_ℓ effects how rapidly the function changes with movement in the ℓ^{th} direction (such as r_1, from the previous parameterisation). In this case, it can be assumed that the output function will be smooth and infinitely differentiable, so p_ℓ can be taken to be equal to 2 – giving a Gaussian covariance [19].

The vector of n observed values is denoted as $\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, also let the $n \times n$ matrix with elements (i,j) given by Equation 9 be denoted as **R**. This means that the likelihood function can be used to find expressions of the optimal mean and variance such as in Equation 10 [19]:

$$\hat{\mu} = \frac{\mathbf{1}'\boldsymbol{R}^{-1}\boldsymbol{y}}{\mathbf{1}'\boldsymbol{R}^{-1}\mathbf{1}}, \qquad \hat{\sigma}^2 = \frac{(\boldsymbol{y}-\mathbf{1}\hat{\mu})'\boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\mu})}{n} \qquad \text{(Equation 10)}$$

Where, $\mathbf{1}$ is a $n \times 1$ vector of ones. This then allows the construction of the 'concentrated log-likelihood function' as shown in Equation 11, a full derivation of which is given by Jones [19]. Note that $|\boldsymbol{R}| = \det(\boldsymbol{R})$.

$$-\frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\boldsymbol{R}|) \qquad \text{(Equation 11)}$$

It should be clear that the concentrated log-likelihood function is dependant only on the values of the kriging parameters ϑ_ℓ and p_ℓ, since p_ℓ has already been set at 2 this leaves only ϑ_ℓ to be found. In practice the concentrated log-likelihood function is maximised to find ϑ_ℓ, this is outlined in Section 3.2.4. To predict values, let the predicted pseudo-observation be ($\mathbf{x}^*$, $\mathbf{y}^*$), also let **r** be the vector of correlations of $Y(\boldsymbol{x}^*)$ with $Y(\boldsymbol{x_i})$ for i=1,... n, such that:

$$\boldsymbol{r} = \begin{pmatrix} Corr[Y(\boldsymbol{x}^*), Y(\boldsymbol{x}_1)] \\ \vdots \\ Corr[Y(\boldsymbol{x}^*), Y(\boldsymbol{x}_n)] \end{pmatrix} \qquad \text{(Equation 12)}$$

The kriging predictor function can then be used to find the predicted output at the pseudo observation, $\mathbf{x}^*$; this function is given in Equation 13. A full derivation of which is provided by Jones [19]:

$$\hat{y}(\boldsymbol{x}^*) = \hat{\mu} + \boldsymbol{r}'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}) \qquad \text{(Equation 13)}$$

As stated previously, each predicted point is treated as a random variable; this means that there is a 'standard error' at each predicted output, given by Equation 14. The standard error has the important property that it is zero at any sampled point, thus this gives a way of checking the software. The standard error is presented in greater detail by Jones [19].

$$s(\boldsymbol{x}^*) = \sqrt{\hat{\sigma}^2\left[1 - \boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{(1-\boldsymbol{r}'\boldsymbol{R}^{-1}\boldsymbol{r})^2}{\mathbf{1}'\boldsymbol{R}^{-1}\mathbf{1}}\right]} \qquad \text{(Equation 14)}$$

Since the standard error can be obtained the probability of reaching a given improvement can also be obtained. Let the target increase to the largest sampled output be denoted as T (for maximising the output function); this then gives the probability of improvement as [19]:

$$P = \Phi\left(\frac{T - \hat{y}(\boldsymbol{x})}{s(\boldsymbol{x})}\right) \qquad \text{(Equation 15)}$$

Where $\Phi(\ )$ is the Normal cumulative distribution function. The exact value of T can change and affect the performance of this optimisation method, although this does mean that T can

be ‘tuned’ to produce a more local, or more global, search. In practice the sample points can be monitored to gauge the locality of the search and adjust T accordingly. The use of a probability to decide the next sample point does produce a very desirable outcome in that the optimum point, when found, can be stated with statistical confidence bounds. For example it could be specified that the optimum is required to be found so that there is a 95% certainty that no other point will produce more than a 5% improvement to the output.

3.2.4. Optimisation of the Kriging Parameters

As part of kriging it is necessary to maximise the concentrated log-likelihood function, Equation 11, by altering the kriging parameters, in particular ϑ_ℓ for each dimension. This means that in attempting to solve a global optimisation problem a second is created, however the function evaluations are cheaper. Once again this is a global optimisation as the output function cannot be assumed to be unimodal. The method used consisted of an initial OLHC sample of possible ϑ_ℓ values and from each of these the steepest gradient direction was evaluated – in a similar manner to the ‘gradient decent’ local optimisation method. As the output function is not known the gradient was approximated as:

$$\nabla f\left(\boldsymbol{\vartheta}^{(\boldsymbol{n})}\right) = \begin{pmatrix} \frac{\partial f}{\partial \vartheta_1} \\ \vdots \\ \frac{\partial f}{\partial \vartheta_d} \end{pmatrix} \approx \begin{pmatrix} \frac{f(\vartheta_1+\varepsilon,\vartheta_2,\dots,\vartheta_d)-f(\vartheta_1,\vartheta_2,\dots,\vartheta_d)}{\varepsilon} \\ \vdots \\ \frac{f(\vartheta_1,\vartheta_2,\dots,\vartheta_d+\varepsilon)-f(\vartheta_1,\vartheta_2,\dots,\vartheta_d)}{\varepsilon} \end{pmatrix} \qquad \text{(Equation 16)}$$

Where, n is the iteration number and d is the number of dimensions being optimised. Then, as per the gradient decent method, the next point is given by:

$$\boldsymbol{\vartheta}^{(n+1)} = \boldsymbol{\vartheta}^{(\boldsymbol{n})} - \lambda \nabla f\left(\boldsymbol{\vartheta}^{(\boldsymbol{n})}\right) \qquad \text{(Equation 17)}$$

A value of λ is then selected to maximise the concentrated log-likelihood function, this is a one dimensional optimisation. The method used to optimise λ was to bracket the maximum and then perform a golden-section search to find the maximum. To ensure the maximum was bracketed the Intermediate Value Theorem (IVT) was used. This is specified as:

$$f'(\lambda_1)\cdot f'(\lambda_2) \le 0 \Longrightarrow \lambda_* \in [\lambda_1,\lambda_2] \;\; such\ that: f'(\lambda_*) = 0 \qquad \text{(Equation 18)}$$

Where $f'(\lambda)$ is approximated as: $f'(\lambda) \approx \frac{f(\lambda+\varepsilon)-f(\lambda)}{\varepsilon}$

Equally, it is clear that a maximum can be found if the stationary point is bracketed and $f'(\lambda_1) > f'(\lambda_2)$ when $\lambda_1 < \lambda_2$. If this is not the case then a minimum or saddle point has been bracketed and therefore can be neglected. Once a maximum has been bracketed, the golden-section search is applied [30]:

$$\lambda_{right}^{(n)} = r\lambda_2^{(n)} + (1-r)\lambda_1^{(n)}; \quad \lambda_{left}^{(n)} = r\lambda_1^{(n)} + (1-r)\lambda_2^{(n)}; \quad r = \frac{-1+\sqrt{5}}{2} \qquad \text{(Equation 19)}$$

If $f\left(\lambda_{left}^{(n)}\right) > f\left(\lambda_{right}^{(n)}\right)$ then $\left[\lambda_1^{(n)}, \lambda_{right}^{(n)}\right]$ brackets the maximum therefore $\lambda_2^{(n+1)} = \lambda_{right}^{(n)}$ and equivalently for the opposite case. This is then iterated until the maximum is found from the golden section search. In order to allow a more global optimisation, the rest of the viable λ range was then considered in the same manner, by stepping the bracket along the possible λ range until bracketing a maximum and then converging to it. This enabled a point which would otherwise converge to a false maximum to move to a region which posed a greater chance of a larger peak. The best value of λ is then used in Equation 17 to find the next ϑ. This whole process then repeats until the maximum is found, this is done for all of the OLHC of initial ϑ values. The value of ϑ which produces the largest value of the concentrated log-likelihood is the one then used in kriging.

Another method of global optimisation is to perform a large enough LHC as to be sure of an optimum value, this can be improved upon by using a smaller LHC but then locally optimising from each point – this is the method followed in this case. Although, the local optimisation could be improved by using the Bi-conjugate Gradient Method rather than a discretised gradient decent coupled to a golden section search, however this was chosen due to the ease of programming. Naturally the benefit of the LHC coupled to a local optimisation is that a 'quality of convergence' can be calculated as the number of points that converged to the same optimum, as a proportion of the total number of starting points. Therefore a high quality suggests that the output function is less fluctuating, whereas a low quality suggests there are a large number of maxima and so the true maximum may have been missed.

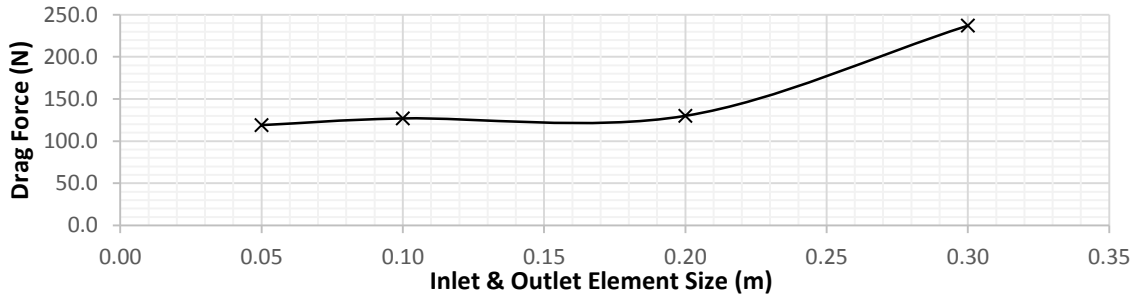
3.2.5. Multidimensional Visualisation

The visualisation of multidimensional sample spaces is an important consideration as it enables the performance of optimisation algorithms to be observed and allows a better understanding of the results. In this case the visualisation was best done by a parallel coordinates plot. This represents each parameter as a vertical axis which is scaled as a proportion of the parameter between 1 and 0. The output data is then grouped; the most appropriate in this case was to split the data into percentiles based on the output, for example the parameter combinations which produced outputs within 10% of the maximum, then the next 10% and so on. These groups were then coloured using an appropriate map to enable an effective visualisation of the results. This was also extended to the probability of improvement and is shown in Figure 10.

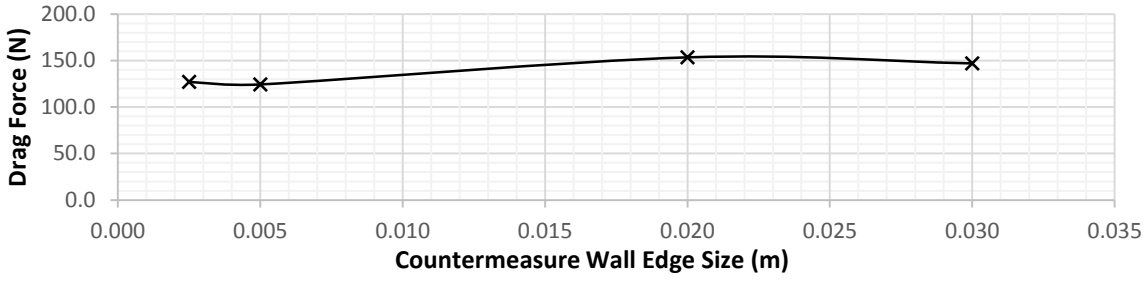
4. Refinement of the Method

4.1. Mesh Convergence

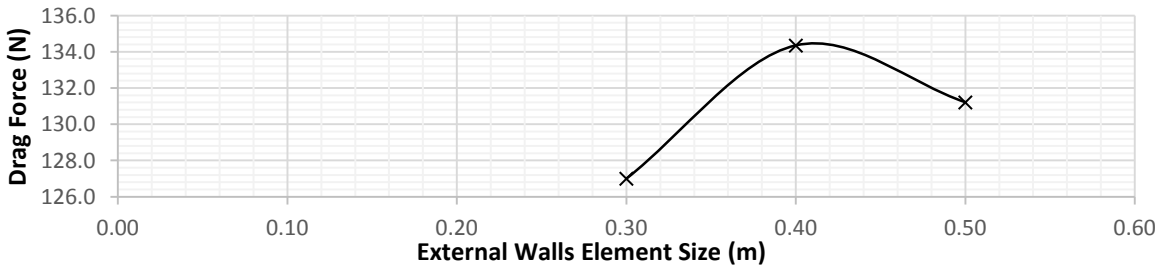
To ensure the results are accurate and mesh-independent it is necessary to perform a mesh convergence study. Due to the complex geometry in this case, advanced sizing functions were used. This means that the mesh convergence study becomes more complex, thus a mesh sensitivity study is performed instead. This is where the parameters which can be modified (in this case, minimum sizes and face sizes) are all adjusted independently of one another, to find the size past which any further refinement produces no change to the result – this is the point whereby the results become mesh independent. The mesh sensitivity study in this case looked at the effect of the cell sizing on the inlet and outlet, the walls of the countermeasure and the external faces alongside the global minimum size. The results of this study are shown in Figure 3 (a) to (d):



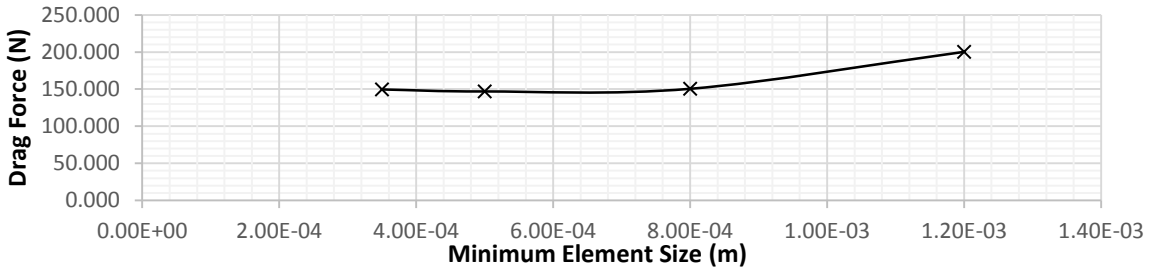
(a)



(b)



(c)



(d)

Figure 3: Mesh sensitivity study results for inlet and outlet element sizes (a), Countermeasure wall edge size (b), External walls element size (c) and the minimum element size (d)

From Figure 3 it is clear that the required mesh sizes are:

Inlet & Outlet: 0.1m
Countermeasure Walls: 0.005m
External Walls: 0.3m
Minimum Size: 5e-4m

For the Inlet & Outlet sizes the 0.1m size has only deviated from the smallest size (0.05m) by 6.7%. Equally, the countermeasure wall size 0.005m only deviates from the 0.0025m size by 2.2%. Despite the appearance of Figure 3 (c) the external wall size only deviates by a maximum of 5.8%, this is marginal and was expected, since the domain was designed so that the external walls are far enough away so that the flow there was mostly undisturbed – allowing the use of far-field boundary conditions. The minimum element size had only deviated from the 3.5e-4m value by 1.8% at the 5e-4m point. This minimal deviation continued past the 8e-4m point; however at the 8e-4m sizing the element quality had decreased and was causing extra mesh smoothing iterations to be required - increasing the computational cost.

4.2. Turbulence Modelling & Inflation Layers

Alongside the convergence to the geometry of the countermeasure it is necessary to refine the mesh in areas where there are rapid changes in variables, in this case the boundary layer. As specified in Section 2.4 this is done with the use an inflation layer mesh defined by the first layer height, the growth rate and the number of layers. The growth rate was taken to be 1.15 so that each layer is 15% thicker than the one before it; this was chosen following research [22]. The number of layers was calculated from the height of the boundary layer using a spreadsheet. The height of the cell layer can easily be calculated and the height of the boundary layer can be approximated from Equation 20 [12]. Therefore the number of layers can be found so that the boundary layer is fully enclosed within the cell layer.

$$\delta = x \cdot \frac{4.64}{\sqrt{Re_x}} \qquad \text{(Equation 20)}$$

Where, δ= height of the boundary layer, x=length of the countermeasure. This was used since the Reynolds number for the countermeasure was found to be 1.55×10^6 and therefore has a

turbulent boundary layer. Finding the first layer height required a mesh refinement study; this was done for both the SA and k-ε, whereby the drag was monitored against the y^+ value at the first layer height. The reason for comparing SA with k-ε is due to the way they handle wall functions – detailed in Section 2.4. The results of this comparison are shown in Figure 4:

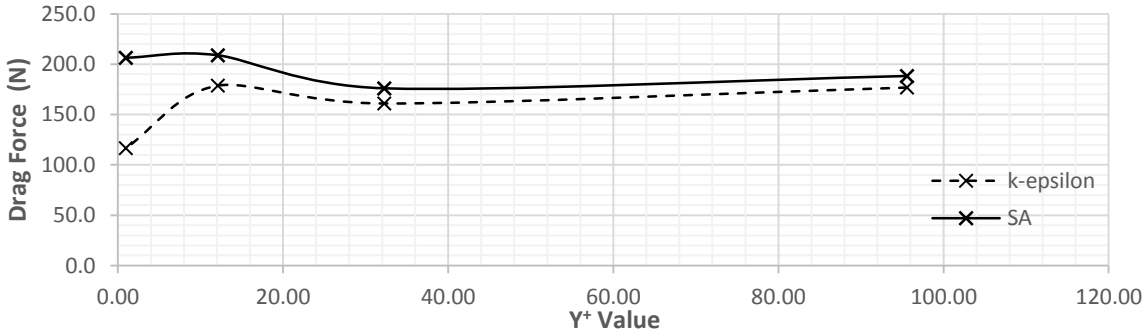


Figure 4: Results of the comparison between the drag forces predicted by the SA and k-ε turbulence models as a function of the y^+ values of the inflation layer mesh.

From Figure 4 it is clear that the SA always predicts a larger drag force than the k-ε model. The optimum operating range for SA is at $y^+ < 2$ [24]; Figure 4 confirms this, as the drag force deviates largely as y^+ increases. The optimum range for k-ε is $y^+>11.225$ up to around 20 [24]; this is again confirmed by Figure 4. The benefit of the k-ε model is that the wall functions can reduce the mesh size. However, it is clear that in this case the model underestimates the drag force and therefore the SA model must be used with a y^+ value of around 1. Although the result will not change greatly up to $y^+=2$, the y^+ value can only be estimated before running the simulation as the local velocity, and therefore wall units, are unknown beforehand. Using $y^+=1$ therefore allows for any local increases in the y^+ values to still remain within the accurate bounds. Were k-ε to be appropriate for use, its drag would be closer to the value from SA – indicating no separation was occurring.

4.3. Simulation Setup

The simulations were set up using a cylindrical domain which was 10 times the diameter of the countermeasure and 2.5 times the length. This geometry was chosen as no changes in the flow occurred at the boundaries except at the outlet. The domain could not have been increased in size due to the computational expense that this would have caused. Although, were greater computational power available, the domain should be increased in length to avoid the wake passing through the outlet as this can introduce some inaccuracy due to insufficient resolution of the pressure gradient behind the countermeasure. The boundary conditions used on the domain were a velocity inlet, pressure outlet and far-field boundaries

on the cylinder sides; the countermeasure had wall boundary conditions. The fin optimisation used the same domain and mesh settings as the trajectory prediction and was run at a mach number of 0.3.

The turbulence model used was the Spalart-Allmaras (SA) with an inflation layer mesh with a $y^+ = 1$, growth rate of 1.15 and 40 layers. This gives a total inflation layer height of 4.2mm compared to a boundary layer height of 1.4mm [12], however there may be local changes in the boundary layer height so the inflation layer has been designed to account for this. The SIMPLE pressure based solver was used with ideal-gas conditions. The ideal gas conditions were required due to the far-field boundaries, however it would also allow for compressibility effects. It is possible that a density based solver would be more appropriate however this was too computationally expensive.

Convergence was assured by monitoring the drag at each iteration, when this levels off the solution has converged. Equally, it is necessary to monitor the residuals to ensure that they are at an appropriate level ($\sim 10^{-4}$-10^{-5}) and that they too have levelled off. This was done in accordance to Section 2.4.

4.4. Software Development

To use the Kriging method of optimisation, a MATLAB program was written. This constructed the OLHC, maximised the log-likelihood function, interpolated the space, evaluated the probability of reaching a target value, identified the three most likely points, wrote the raw data to text files and produced two parallel coordinates plot of the output and the probability alongside the sampled points. Although a number of programs which perform kriging, or another method of optimisation, are available it was believed that, by producing one, a greater level of understanding would be achieved alongside being more customisable to this specific case. This program is available from the author on request.

A number of issues were realised in writing the software, this caused the method to evolve. Firstly it was found that A is limited by the diameter of the projectile and r_2 and therefore it is better to express A as a proportion of the maximum that it can be. This was not realised as the best course of action until after the OLHC of initial simulations had been run, in order to save on computational cost the OLHC was not re-run. This meant that A was heavily sampled at 1 however the kriging model would still be able to find a maximum at other points due to the method in which it operates. Equally, due to the properties of the OLHC if one dimension is

collapsed the remaining dimensions are still orthogonal, this means that the other dimensions are equally well represented and that only sampling in A has been affected.

The interpolation of the output values by the kriging model was computationally cheap however the total number of points in the space is $1.2x10^8$ this would take too long to run for general use and so a Latin Hypercube (LHC) was used to sample the space. The total number of points to be sampled had to be decided upon; to do this the model was run using $4x10^4$, $4x10^5$ and $1.2x10^8$ sample points. All of these runs gave similar points for the most probable increases in the output value and so it was decided to use a LHC of $4x10^4$ points.

The input values were normalised using a Euclidean norm such that they form a unit vector, this is done to remove dimensional dependence from the variables when they are used in kriging. The output values are then multiplied by the norm of the input data to scale them accordingly. The optimisation index was scaled by the maximum of the OLHC sample for ease of comparison.

Alongside the main kriging program for the fin optimisation, a second was written for the kriging interpolation of the drag coefficients, this is a simpler case as it is only 1 dimensional and without optimisation requirement. However, the standard error was evaluated to give a measure of the accuracy of the interpolation at each point.

5. Results & Discussion

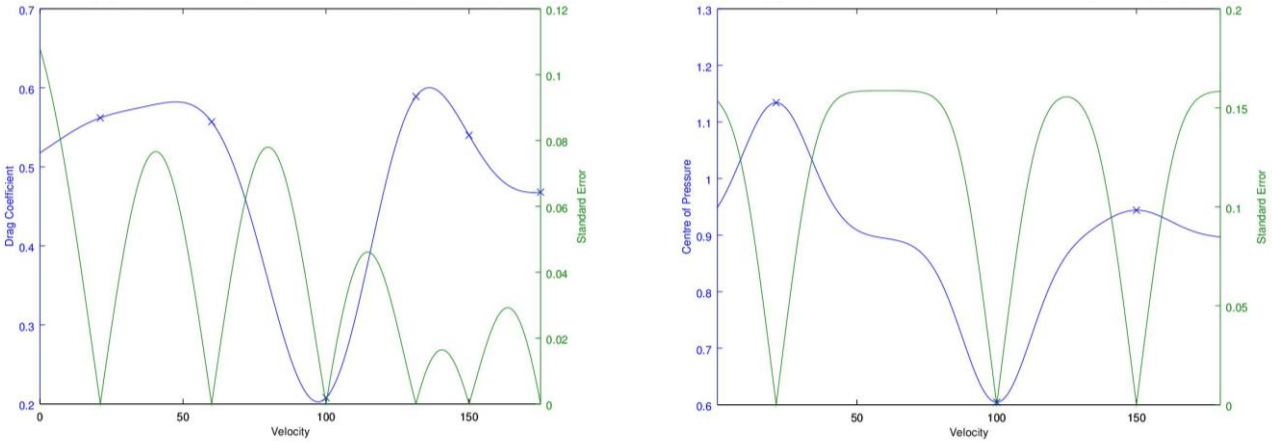
5.1. Time-stepping and Trajectory

5.1.1. Drag Coefficient and Centre of Pressure

The CFD analysis was used to calculate the drag coefficient at a number of velocities; these were then interpolated using kriging, as explained in Section 3.1.4. The resulting relationship between the drag coefficients and the velocity is shown in Figure 5, with the sampled points from CFD given in Table 3.

Table 3: Sampled velocities and the CFD calculated drag forces

Z-Velocity (ms^{-1})	Drag Force (N)	Drag Coefficient
21	2.04	0.562
60	16.35	0.557
100	16.98	0.208
131	82.97	0.589
150	99.01	0.540
175	116.97	0.468



(a) **(b)**

Figure 5: Kriging interpolated drag coefficient (a) and Centre of Pressure (b) as a function of the velocity (m/s) in the z direction, with the kriging standard error.

The interpolated values for the drag coefficient were then used alongside the time-stepping to predict the trajectory, the results for which are given in Sections 5.1.2.–5.1.4. Generally the drag coefficient is around 0.5-0.6, which is lower than the value used by Chemring of 0.8 - a value found using analytical approximations. There is a large deviation from the trend of the drag coefficient at 100m/s, where it drops to around 0.2. It is expected that this could be due to a drag crisis, although whether this would be over the whole countermeasure or on a specific surface, such as the fins, required further investigation. It could also be that the simulation at 100m/s was incorrect and therefore caused an anomalous value, thus further investigation was performed. The standard error remains below 0.1 for most of the values this suggests that the interpolation is fairly accurate although adding in more sample points will reduce this.

The centre of pressure (CoP) was also calculated as it was required for the three d.o.f case. However, this was only done for three of the sampled velocities as the three degree of freedom case was originally an extension to the project and so it was not initially being investigated. Furthermore it had been assumed that the CoP was unlikely to change with the velocity so it was found at the initial velocity of 21m/s and at 150m/s to confirm this. However, the rapid drop in the drag coefficient at 100m/s later suggested that the centre of pressure should be calculated there as well. Since the simulations were run in parallel across multiple computers, some of them converged faster than others and so were post-processed sooner. The 100m/s case took longer to converge and so it was not known that it had a large deviation until after the other simulations had been post-processed. This delayed convergence may well have been due to a drag crisis for which it is expected that it would take longer to

fully develop the turbulence. The CoP was interpolated using kriging and is shown in Figure 5 (b) where the CoP is being measured from the nose of the countermeasure.

The standard error in this case is larger due to the infrequent sampling; this should be improved upon by sampling more velocities. It is clear that at 100m/s the CoP moves forwards towards the nose. This would be consistent with the hypothesis of a drag crisis occurring at the fins. As the local drag of the fins decreases, the balance of forces on the countermeasure will change so that the majority of the force will be acting at the nose – hence the CoP moves forwards. Due to this support for the theory of the drag crisis at the fins as well as the use of correct simulation practice with mesh convergence studies and the residual and output monitoring, it can be assumed that the simulation at 100m/s is accurate. The CoP values can then be used in the three d.o.f case to find the changes in angle α.

5.1.2. One Degree of Freedom

The one d.o.f case only considers movement in the z direction, therefore the angle, α, remains constant at 45°. This case resulted in a trajectory as shown alongside the other cases in Figure 9, this case is unrealistic as there will be movement in the other degrees of freedom, r & α, however this case was the initial case performed as it is the simplest and tests the concept.

The velocity calculated in this case can be tested against the velocity data provided by Chemring, as done in Figure 6. This is an important comparison as it shows that the velocity calculated by this project follows a very similar development over time as the data used by Chemring, which has been seen to be similar to the true values in tests of the countermeasures. The minor deviation from Chemring's data could be due to the different time-stepping schemes since Chemring use an RK4 whereas this project used an AB5-AM5 predictor-corrector scheme. However, it could also be due to the difference in the drag coefficients used. It is worth noting that the data from Chemring was only provided up to two seconds into flight.

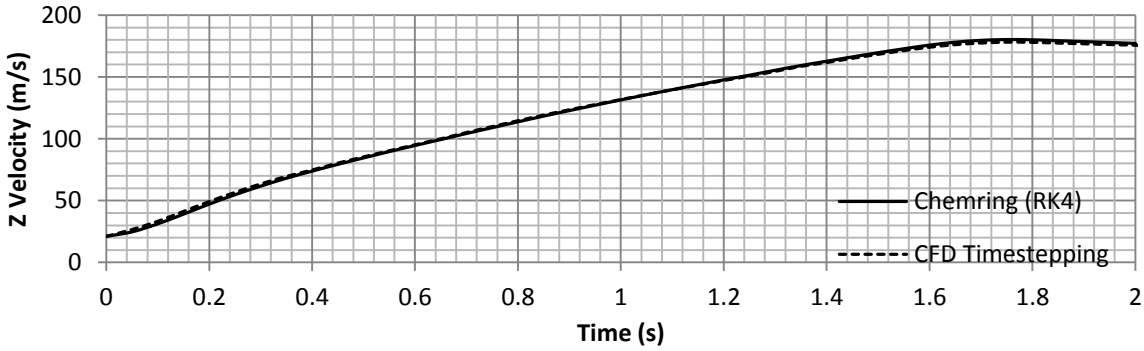


Figure 6: The velocity (m/s) as a function of the time (s), for both the CFD time-stepping and the data from Chemring

5.1.3. Two Degree of Freedom

The case for two d.o.f considers both z and r velocities, this gives the velocity profiles over the 4.4s flight of the countermeasure as in Figure 7. The thrust occurs over the first 1.8s, the exact values for which were provided by Chemring. The z velocity follows the same profile as in Figure 6 due to an assumed independence between z and r – as explained in Section 3.1.3. The profile of r results from acceleration due to gravity being countered by a velocity dependent drag force, although the result is an almost linear relationship.

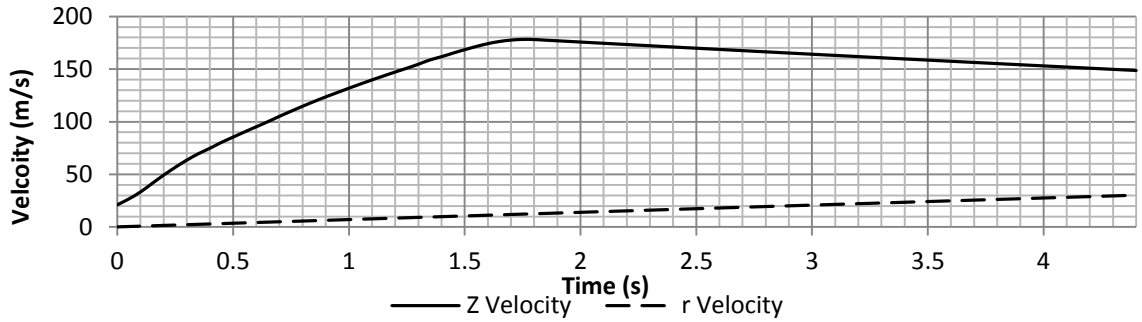


Figure 7: The velocity (m/s) as a function of the time (s), for both the z and r directions

The assumed independence between z and r was, in part, justified by the assumption that the velocity in the z direction would be significantly higher than in the r direction. Figure 7 shows that this assumption was correct, although whether this enables the assumption of independence is still debateable and should be investigated further following this project.

5.1.4. Three Degree of Freedom

The most full trajectory model that this project produced was in three d.o.f (z, r & α). It is worth noting that the CFD analysis was only performed considering the z direction with analytical values for the r direction – as explained in Section 3.1.3. The velocities calculated in each d.o.f are shown in Figure 8; from which it is clear that the r velocity now has a more parabolic profile whereas the z direction velocity maintains a similar profile to the previous cases. The velocity in α increases later in the flight and is almost proportional to the integral of the r velocity, a likely occurrence as it is the drag force in the r direction which is the main cause of the rotation and is dependent on the r velocity.

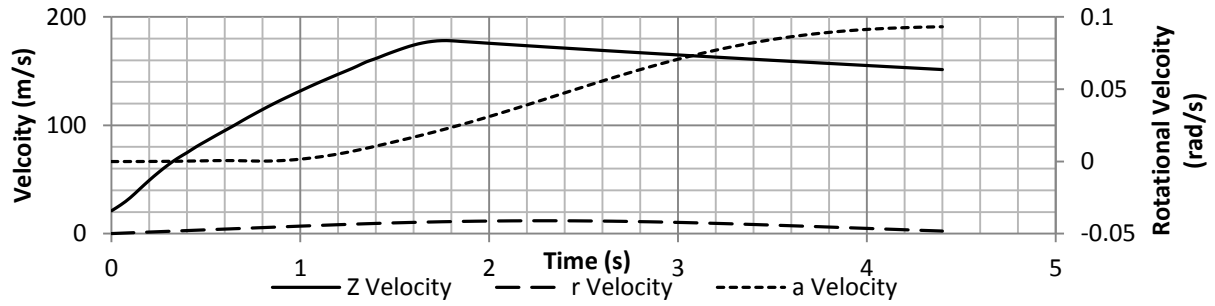


Figure 8: The velocity (m/s) as a function of the time (s), for both the z and r directions

The trajectories produced by each of the cases investigated are shown in Figure 9, where the crosses mark the point at which the drogue chute will be deployed, ending the flight after 4.4s from the start. It is clear that the 1 d.o.f and 2 d.o.f cases both overestimate the height achieved but underestimate the horizontal distance.

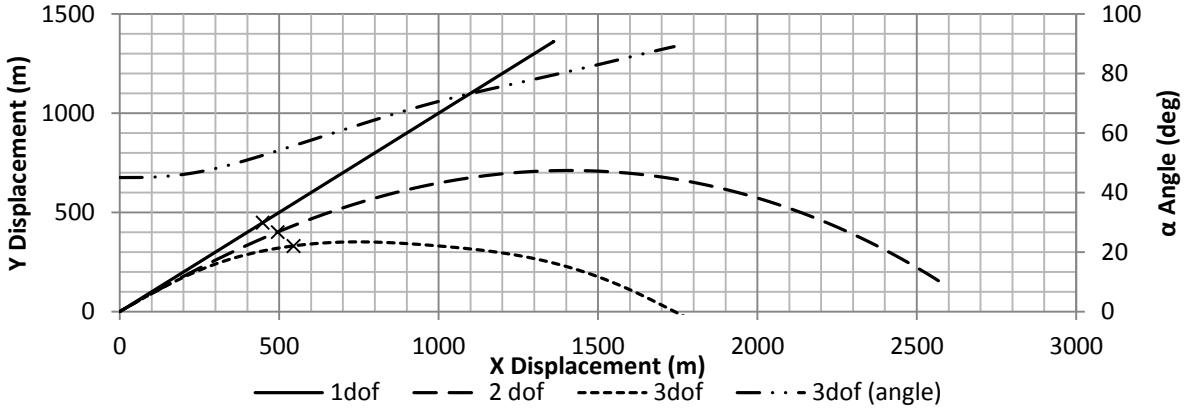


Figure 9: The Trajectory of the countermeasure when considering 1, 2 & 3 degrees of freedom, including the angle, α, for the 3 d.o.f case.

Also worth noting from the 3 d.o.f case is that the angle of the countermeasure is not at the tangent to the trajectory, as was expected. This suggests that the countermeasure does not fly fully nose first, particularly if the flight time were to be extended to beyond the peak height. The cause of this non-tangential angle would be that the restoring force, in r, does not produce enough torque to correct the angle quickly, consequently suggesting that an improvement in the fins could be made such that the tail would be brought in line with the flight direction of the countermeasure. This could be done by using a symmetrical aerofoil profile on the fins, as these would provide a large amount of restoring force without jeopardising the low zero-angle drag force.

It is worth noting, however, that due to the assumed independence of z and r and the use of analytical values for r, which assume it is an infinite cylinder, the effect of the fins would not be well represented and to be sure of an accurate representation it would be worth running the full 3 d.o.f model in CFD.

5.1.5. Accuracy

The simulations were performed using monitors on the outputs and residuals, appropriate boundary conditions and a fully converged mesh, see Section 4.1. Alongside this the boundary layer was well represented, see Section 4.2, and the y^+ values were checked after the simulations to ensure that they were all below 2, the threshold for accurate boundary layer representation using the SA turbulence model. Alongside this, the eddy viscosity ratio in the boundary layer was checked to ensure that the inflation layer of cells fully captured the boundary layer. Although plots of the residuals and output parameters against the number of iterations were obtained following the simulations they have not been included for conciseness but are available from the author upon request. These measures suggest that the simulations themselves were accurate; these practices were applied to all simulations run as part of this project.

The accuracy of the time-stepping method itself is very high, since the error is of the order of Δt^5 with $\Delta t=0.05s$ this means that very little error will be present over the 4.4s flight time. The predictor-corrector scheme used removes instability so that the error will not grow over time, therefore it will remain very small. The accuracy of the drag coefficient should be high as a reasonable number of points were sampled as well as the kriging interpolation being very accurate. The centre of pressure will be less accurate due to fewer sample points. Both of these could be improved in accuracy by a larger number of sample points.

A number of assumptions were made to simplify the trajectory prediction, these were necessary as without them the prediction could only have been achieved by a full 3 d.o.f model in CFD, which would have been too computationally expensive and beyond the scope of this project. The main assumption made was that the three d.o.fs were all independent, however velocity in the r direction could increase separation or turbulence and therefore the drag force in the z direction. It is difficult to assess the magnitude of the inaccuracy this would introduce. However, since the velocity in the r direction is very minimal compared to the velocity in the z direction, it is believed that this inaccuracy would be small as separation would be unlikely and therefore the overall effect this would have on the trajectory should be equally small. To ascertain the magnitude of this inaccuracy and therefore whether the assumption is valid it would be necessary to run a full 3 d.o.f CFD model as a comparison.

Another assumption made was that the countermeasure was equivalent to an infinite cylinder to find the drag in the r direction – this neglects the effects of the fins and countermeasure

ends. Therefore the inaccuracy of this could be out by a large proportion this would affect the angle most dramatically with a minor consequence on the rest of the trajectory. Again this would be best evaluated by a full 3 d.o.f CFD model.

The overall accuracy of the trajectory prediction was within expected bounds. In particular, the drag coefficient was marginally lower than the analytical value used by Chemring and the trajectory was within the region of observed behaviour of the flight of the countermeasure – thus paying credence to the accuracy of this method.

5.2. Fin Optimisation

5.2.1. Orthogonal Latin Hypercube Sample

The OLHC sample of simulations produced data for 16 cases of fin design. The optimisation index was taken to be $\frac{Moment}{Force^2}$ this puts extra emphasis on the drag force, this was then normalised to remove dimensionality. Other indices could have been used and this remains the main source of ambiguity in this optimisation method; ambiguity could be removed by setting target values set for the moment or drag force, such as to spin at a given speed. The values of the optimisation index predicted by the kriging model are shown in Figure 10 (a), which produces a probability of improvement as shown in Figure 10 (b). The lines shown in black are the sampled points. It is clear that the probability of improvement has a number of possible maxima, thus the next iteration can sample the 2 most probable points to find the optimum faster. The collapse in A sampling, discussed in Section 4.4, is also clear.

5.2.2. Iterations

Three iterations of the method were performed, however the first included two designs so a total of 20 fin designs fin were tested. The parameter combinations resulting from the iterations are given in Table 4 alongside the probability of improvement.

Table 4: Parameter combinations from the fin optimisation, with probabilities of improvement

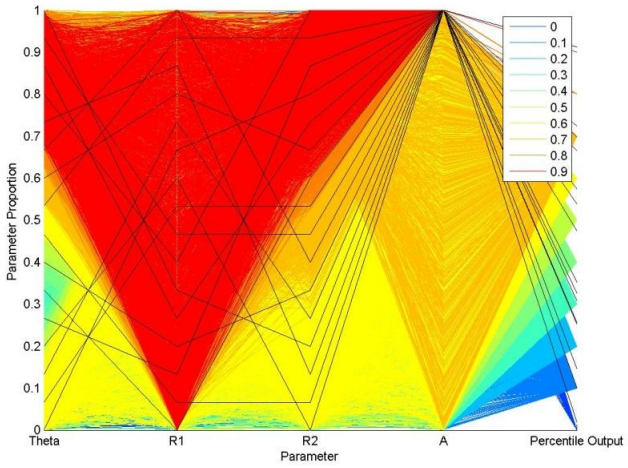
Iteration	Design Number	Parameter Values				Probability of improvement	Optimisation Index	
		θ (deg)	r_1 (m)	r_2 (m)	A (deg)		Actual	Predicted
0	**17**	309	0.007	0.98	89	0.048	0.243	0.95
	18	314	0.048	1.00	86	0.047	0.931	0.95
1	***19***	317	0.018	1.00	17	0.040	0.569	0.88
2	***20***	220	0.019	0.79	20	0.023	0.833	0.82
3	**21**	321	0.018	0.68	21	0.020	n/a	0.82

It is clear from Figure 10 that the probability of improvement forms a distinct set of paths across the plots which represent the maxima. However, the magnitude of the probability compared to previous iterations is not apparent from Figure 10 although from Table 4 the

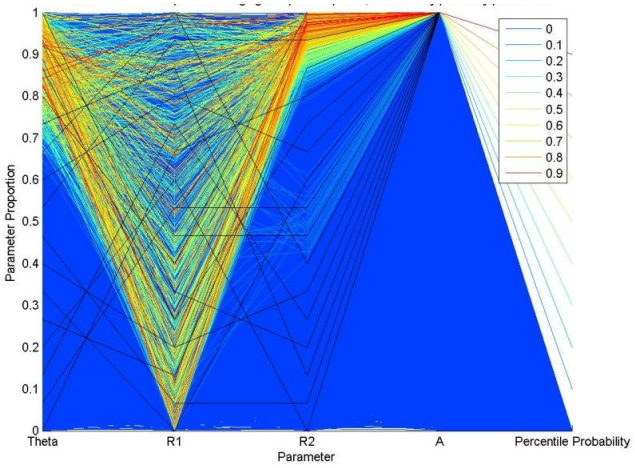
probability is seen to decrease with each iteration, which demonstrates the increasing accuracy of the kriging model. The purpose of the fin optimisation was to demonstrate the method rather than to find a general optimum design, since the best design will depend on constraints unique to purpose, thus only three iterations were performed. Jones [19] demonstrates the method for a 1D case using 11 iterations for the whole optimisation. This was not possible in this case due to computational expense, however the effectiveness of the method is clear as the probability of improvement has been narrowed down to only a 2% chance of a 25% improvement. The unique aspect of this method is that the optimum can be parameterised in this way, whereby a statistical significance can be set along with the maximum improvement. This means that, in this case, it can be stated that, no other design will achieve more than 25% improvement with a certainty of 98%.

Iter

0

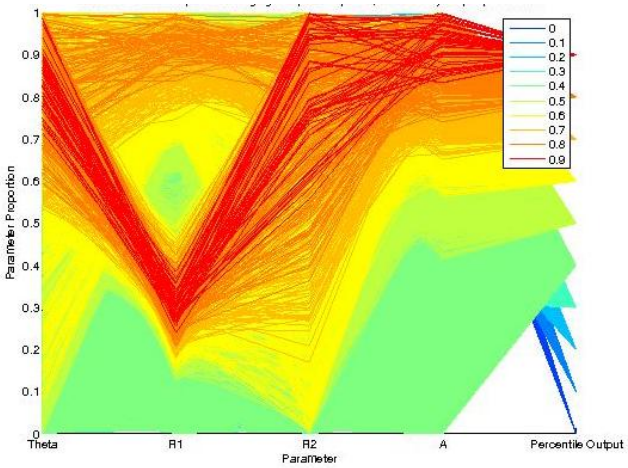


(a)

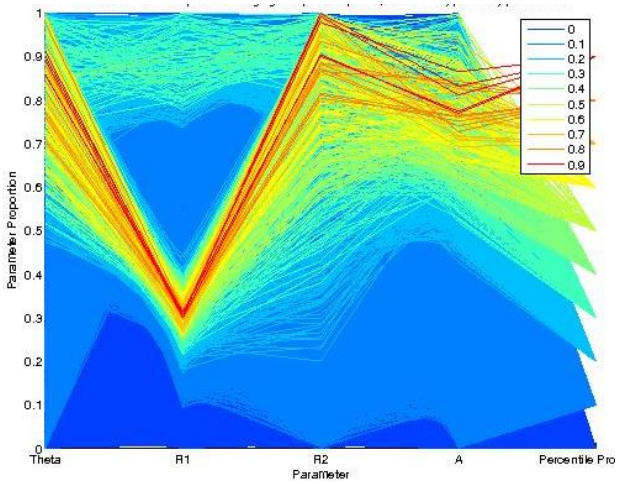


(b)

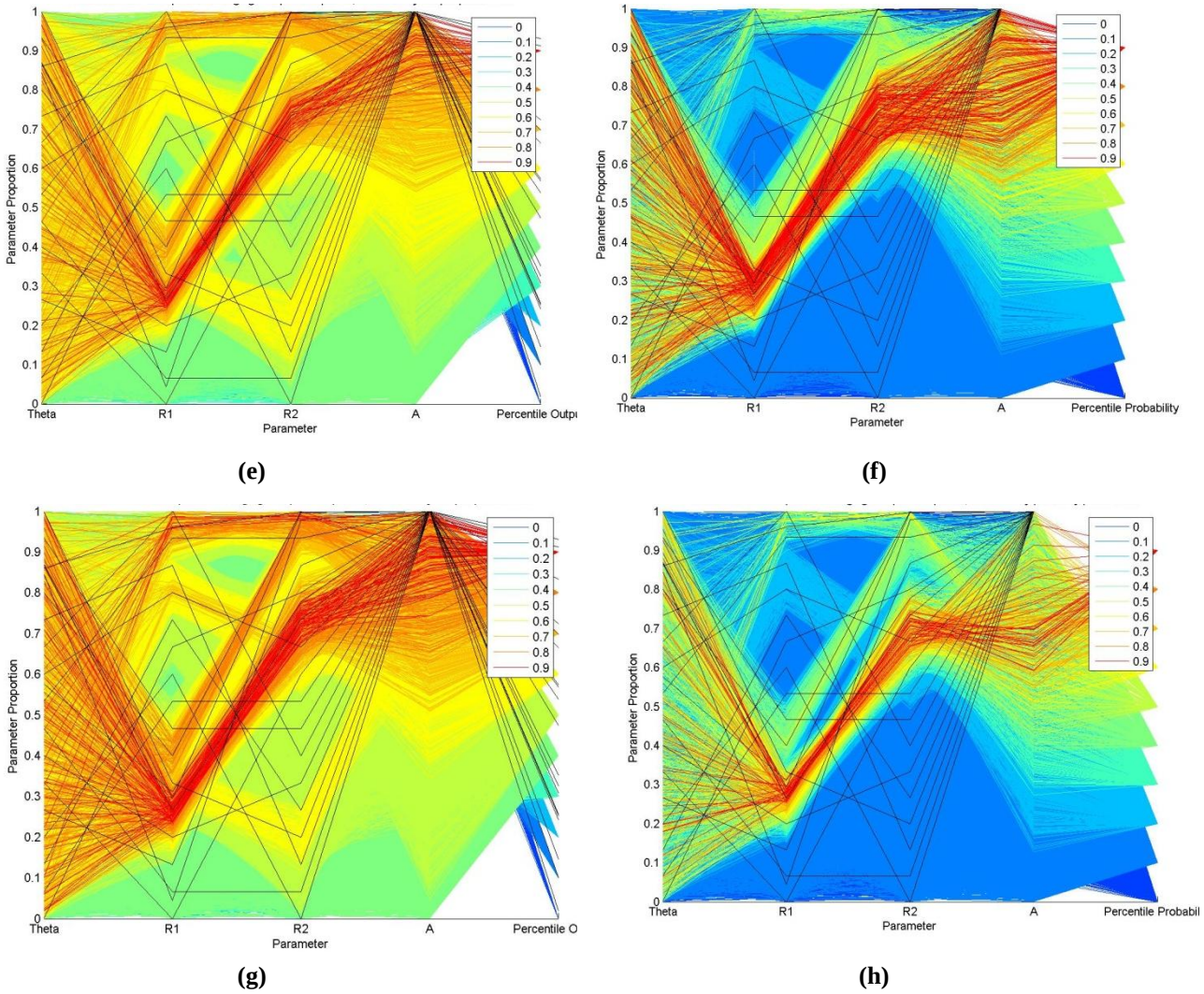
1



(c)



(d)



(e) **(f)**

(g) **(h)**

Figure 10: Parallel coordinate plots of the optimisation index (a), (c), (e) & (g) and the probability of a 25% improvement on the highest sample point (b), (d), (f) & (h), as predicted by kriging, for each iteration of the optimisation process.

The ambiguity of the next design has been removed by this method, so can be programmed into the CFD software to find the optimum. There may be some inaccuracy in this method as the kriging sample across the space cannot sample all the points, due to the computational expense. However, the sample quantity was found to be large enough to represent it effectively, as explained in Section 4.4. The inaccuracy therefore should be small and should not affect the outcome of the optimisation process but may affect its efficiency. There may be some inaccuracy in the simulations however this should have been removed by following the practices discussed in Section 2.4. Since the geometry is very similar to the countermeasure provided by Chemring, the same mesh settings were used. Furthermore, any inaccuracies are likely to be systematic and therefore affect all the designs equally. In this way some inaccuracy could be adopted to save on computational cost, such as running using a coarse mesh to find likely designs and then running them with a finer mesh.

The best design achieved by this optimisation process was design 14, part of the initial OLHC sample, this design is shown in Figure 11. It is clear from Figure 11 that the design is fairly effective as there is a distinct stream of flow being directed to produce a moment. The design is also somewhat intuitive and has a balance of both a streamlined profile and the rotation required for torque generation, suggesting that the simulations have been accurate and produced a reliable result. The results for each design are shown in Table 5, with each optimisation index. It is clear that design 14 produced the best result with a few designs producing results very close to it.

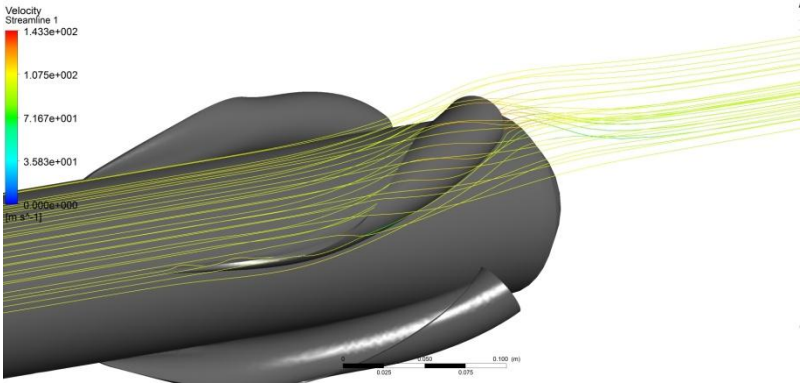


Figure 11: Streamlines coloured by velocity for design 14, the best performing design achieved by the optimisation.

Table 5: Design results from the fin optimisation with optimisation indices

Design Number	Theta (deg)	r1 (m)	r2 (m)	A (deg)	Force (N)	Torque (Nm)	Index (mN^{-1})	Index (normalised)
1	30	0.035	0.81	21	42.33	0.05	2.93E-05	0.02
2	52	0.029	0.56	26	40.51	0.25	1.49E-04	0.10
3	74	0.038	0.30	16	46.95	1.86	8.44E-04	0.57
4	96	0.032	0.05	62	58.01	2.35	6.99E-04	0.48
5	118	0.011	0.87	20	36.08	0.59	4.53E-04	0.31
6	140	0.005	0.62	24	38.61	0.56	3.77E-04	0.26
7	162	0.014	0.37	5	43.39	0.91	4.81E-04	0.33
8	184	0.008	0.11	50	35.97	0.28	2.15E-04	0.15
9	206	0.047	0.94	20	157.93	25.63	1.03E-03	0.70
10	228	0.041	0.68	23	45.55	2.21	1.06E-03	0.72
11	250	0.050	0.43	22	159.91	20.27	7.93E-04	0.54
12	272	0.044	0.18	46	222.92	18.50	3.72E-04	0.25
13	294	0.023	1.00	19	69.34	6.45	1.34E-03	0.91
14	**316**	**0.017**	**0.75**	**22**	**53.19**	**4.16**	**1.47E-03**	**1.00**
15	338	0.026	0.49	11	77.07	3.08	5.18E-04	0.35
16	360	0.020	0.24	40	84.83	7.05	9.79E-04	0.67
17	309	0.007	0.98	89	41.68	0.62	3.57E-04	0.24
18	314	0.048	1.00	86	125.31	21.51	1.37E-03	0.93
19	317	0.018	1.00	17	53.00	2.35	8.38E-04	0.57
20	220	0.019	0.79	20	60.90	4.54	1.23E-03	0.83

5.2.3. Accuracy

The accuracy of simulations will be as discussed in Section 5.1.5. Although it is worth noting that the CFD model did not account for the axial rotation of the countermeasure and so the fin design would be the optimum for initial acceleration. The best design for the whole trajectory should take into account this rotation and include the angle of the leading edge of the fin as a design parameter.

The accuracy of the optimisation method is more dependent on the kriging model. The model itself is very accurate and has been widely used. However, the optimisation of the kriging parameters in this case was not necessarily robust although after a number of test runs and calibrations the optimisation gave the same results for the parameters regardless of the adjustments within the program. Thus the method has a reasonable level of robustness for a global optimisation method, since all global optimisation methods will have a level of inaccuracy due to the inability to sample all points. The inaccuracy of the kriging model will not affect the optimum design but the efficiency of the method and bounds of the optimum. Thus in this case there may be a higher probability of improvement than the 2% it was narrowed down to, although the exact level of inaccuracy in this cannot be established. However, it is worth noting that the kriging model predicts some of the fin design iterations (18 & 20) with an error of around 2.1%, although design 17 had an error of 74.4%, see Table 4. This is expected since the method targets areas with a high standard error as well as those with a high mean by using the probability of improvement; this gives some indication to the accuracy of the kriging model. Improvements to the accuracy would be to use a more robust global optimisation method for optimising the kriging parameters, such as a large LHC coupled to the bi-conjugate gradient decent method.

6. Conclusions

6.1. Time-stepping and Trajectory

This project successfully confirmed the viability of using CFD alongside an AB-AM time-stepping scheme to predict the trajectory. This provides the basis for future projects to use these schemes and the simulation setups to program a full 3 d.o.f CFD program which automates the whole process. The trajectory prediction demonstrated in this report is both accurate and provides a large amount of useful information, for example it would be possible to notice if the countermeasure started to tumble or to not fly fully nose first in which case recommendations to improve the design could be given such as refining the fins. Alongside

this ability to present full in-flight behaviour and enable identification of required design changes, the trajectory prediction method would be easily adapted for changes in initial launch conditions or weather conditions such as a side or head wind. Were the method to be programmed into a CFD model to give a more accurate prediction, the additional changes in conditions could be easily added in.

6.2. Fin Optimisation

The fin optimisation resulted in design 14 being the best design found with a 98% certainty that no other design could achieve more than a 25% increase in the optimisation index. However the design itself is secondary to the method used, which has been found to be very effective in that it removes ambiguity in the optimisation process and narrows down a multi-parameter optimisation problem very efficiently. The kriging optimisation method is easily expanded to any number of dimensions; this means that it can be applied to any optimisation problem that can be expressed as a number of changeable parameters and an optimisation index.

Further research should focus on improving the efficiency and robustness of the method and extending its application. In particular, the method should be combined with a full 4 d.o.f CFD model of trajectory prediction, which takes into account axial rotation in β, to produce a CFD program which is capable of taking countermeasure geometries alongside thrust profiles, weather conditions, launch conditions, variable parameters and the optimisation objectives to then step through the whole flight and find the optimum design for the task. This could include optimisation of the body shape and nose cones for example and could be extended to other projectile based applications such as missiles, rockets or other countermeasures. This combination of the two methods investigated in this report would be easily developed from this research and would produce a very powerful piece of design software which would enable a greater survival rate of the platform, by improving the design, as well as reducing the cost of design development, by reducing the need for experimental testing.

7. Project Management, Sustainability and Risk

The management of the project was monitored by the Gantt chart. A logbook was also being maintained with the work done being reviewed weekly; extra work was added into the following week, should any be incomplete due to unforeseen difficulties. Regular meetings with the supervisor were attended to monitor progress with any extra meetings being

scheduled as required. The project began as a number of base tasks with a series of extensions which could be added on, these were still in keeping with the initial aims but produced a more in depth result. The extensions were worked on when it was deemed that the project was ahead of schedule or could be easily expanded without much delay to the current work. This mitigated any difficulties which could have been experienced by having a base work which was substantial but easily completed within the time. Then, were fewer difficulties encountered than expected, the extensions were added in.

The sustainability aspect of this project is more focussed on the effects that the optimisation aspect could have on improving efficiency. This project investigated optimisation which could be extended to other applications, which would reduce the waste energy and improve efficiency. This is an important aspect to consider as part of sustainability, as it does not require any change to current lifestyle choices, which are harder to alter, but instead alters the current technology to become more sustainable by reducing the energy usage. This means that it could drop below a sustainable level of energy use. It is not possible in this case to investigate the environmental risks and impacts of the countermeasure due to confidentiality limitations. However, the simulations use a large amount of energy in running, this is difficult to mitigate but correct simulation practices reduce the computational expense and therefore the energy usage. The use of renewable, or more sustainable, energy sources would mitigate the electricity used in the simulations, although they are not widely used at the moment this is a field that the research presented in this project could impact heavily on, enabling a wider usage and more sustainable production of energy across the globe.

The health and safety risks of this project are minimal as it is research based. However, there are a number of necessary practices required when working for long periods of time at computers. For example, it is necessary to have the desk, monitor and chair at the correct heights to prevent back or neck strain. It is also necessary to take regular breaks and to prevent glare on the screen of the computer. In order to follow these, the same computers were used and set up to the correct positions at each time of usage. Equally, any computers at which the glare could not be avoided were not used and were reported to the appropriate people. In this way any risk of a repetitive strain injury, back and neck problems or harm to vision were mitigated and no issues were encountered.

Table 6: Time management scheme provided in the form of a Gantt chart

	Week:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	Tasks (* = milestone) | Commencing:	12/01	19/01	26/01	02/02	09/02	16/02	23/02	02/03	09/03	16/03	23/03	30/03	06/04	13/04	20/04	27/04
1	Import Chemring geometry into CFD software																
2	Run test simulation		*														
3	Mesh convergence study																
4	Set up timestepping spreadsheet/program																
5	Timestep for 1 d.o.f				*												
6	Compare with known data																
7	Check results for computational accuracy & correct as needed																
8	Add in extra d.o.fs one at a time (repeating checks)							*									
9	Compile all d.o.fs in flight to model behaviour against time								*								
10	Finish kriging program		*														
11	Run test simulation on fin																
12	Run initial OLHC sample simulations					*											
13	Check results for computational accuracy and correct as needed																
14	Use kriging algorithm to find optimum (checking each simulation)							*									
15	*Compare flight of optimum fin projectile against initial design*																
16	*Find flight behaviour at different launch conditions*																
17	Final report																*

Table 7: Project risk assessment

ID	Risk item	Effect	Cause	Likelihood	Severity	Importance	Action to minimise risk
1	Confidentiality agreement delay	Time delays to the access of data and geometry, may affect time available for other tasks.	Required cooperation from both the company and the university, agreements must first be compiled.	4	3	12	Agreement prepared in 1st term and to be signed off over Christmas break. Once signed off, data and geometry will be requested as soon as possible. There is the option for fin optimisation being commenced before projectile flight analysis.
2	CFD run time	Delays to the project occur, less time available for other tasks.	Simulations take a long time to run or mesh.	5	2	10	Simultaneous running of fin optimisation and projectile analysis. Multi-processor computers could be used to speed up the run time.
3	CFD setup errors	The simulation does not get accurate results, delays occur to other tasks.	Complex flow conditions and geometry.	3	3	9	Seek advice from supervisors as to the correct setup. Run other simulations for other parts of the project whilst setting up the complex simulations.
4	Failure of time stepping	The flight behaviour can only be found for a few known points. A complete analysis is not possible.	Complex time stepping algorithms, possible computational instability.	2	2	4	Run the simulation at the known data points provided by Chemring, thus reducing analysis of the projectile flight to only few points.
5	Failure to run all degrees of freedom	Only some degrees of freedom can be analysed so the projectile behaviour will not be complete.	Complex flow in multiple d.o.fs.	4	2	8	Only run the simulation taking into account a few degrees of freedom, thus reducing the analysis of the projectile flight to a partial analysis.
6	Failure to run optimisation algorithm	Only approximate manual optimisation can be performed.	Complex optimisation algorithms and surrogate models.	3	3	9	Performing manual optimisation by adjusting the parameters and iteratively approximating an optimum point.

References

[1] **Despeyroux, Antoine, et al.** *Numerical analysis of static and dynamic performances of grid fin controlled missiles.* Quebec City : American Institue of Aeronautics and Astronautics, 2014.

[2] *Flight Performance of a Small Diameter Munition with a Rotating Wing Actuator.* **Fresconi, Frank, DeSpirito, James and Celmins, Ilmars.** 2014, Journal of Spacecraft and Rockets, pp. 1-15.

[3] *Control Performance, Aerodynamic Modeling and Validation of Coupled Simulation Techniques for Guided Projectile Roll Dynamics.* **Sahu, Jubaraj, Fresconi, Frank and Heavey, Karen R.** 2014, American Institue of Aeronautics and Astronautics.

[4] *Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics.* **Baysal, Oktay and Eleshaky, Mohamed E.** 1992, American Institute of Aeronautics and Astronautics, pp. 718-725.

[5] *Numerical Prediction of Pitch Damping Stability Derivatives for Finned Projectiles.* **Bhagwandin, Vishal A and Sahu, Jubaraj.** 2014, Journal of Spacecraft and Rockets, pp. 1603-1618.

[6] *Navier-Stokes Predictions of Dynamic Stability Derivatives: Evaluation of Steady-State Methods.* **DeSpirito, James, Silton, Sidra I and Weinacht, Paul.** 2009, Journal of Spacecraft and Rockets, pp. 1142-1154.

[7] *Ramjet Powered Missile Design Using a Genetic Algorithm.* **Hartfield, Roy J, Jenkins, Rhonald M and Burkhalter, John E.** 2007, Journal of Computing and Information Science in Engineering, pp. 167-173.

[8] *Aerodynamic shape optimization techniques based on control theory.* **Jameson, Antony, et al.** 1998, American Institute of Aeronautics and Astronautics, pp. 1988-2538.

[9] *Numerical Experiments on Finned Bodies.* **Silton, Sidra I.** 2014, American Institute of Aeronautics and Astronautics.

[10] *Applications of computational fluid dynamics to the aerodynamics of army projectiles.* **Sturek, Walter B, et al.** 1994, Journal of Spacecraft and Rockets, pp. 186-199.

[11] **Sigal, Asher.** *Analysis of the roll moments of finned missiles.* Haifa : American institue of Aeronautics and Astronautics, 1997.

[12] **Tabor, Gavin.** Formula Sheet. *Exeter Learning Environment.* [Online] 20 10 2009. [Cited: 10 12 2014.] http://vle.exeter.ac.uk/pluginfile.php/119843/mod_resource/content/0/ECM3110-1/fluidsStudyResources/FormulaSheetExam.pdf.

[13] **Benson, Tom.** Speed of Sound. *National Aeronautics and Space Administration.* [Online] 12 06 2014. [Cited: 01 04 2015.] http://www.grc.nasa.gov/WWW/k-12/airplane/sound.html.

[14] **CFD-Online.** Subsonic Flow. *CFD-Online.* [Online] 27 09 2005. [Cited: 01 04 2015.] http://www.cfd-online.com/Wiki/Subsonic_flow.

[15] **Benson, Tom.** Centre of Pressure - cp. *NASA Glenn Research Centre.* [Online] 12 06 2014. [Cited: 02 11 2014.] http://www.grc.nasa.gov/WWW/k-12/airplane/cp.html.

[16] *A construction method for orthogonal Latin hypercube designs.* **Steinberg, David M and Lin, Dennis K J.** 2006, Biometrika, pp. 279-288.

[17] *Optimization using surrogate models and.* **Forrester, Alexander I J, Bressloff, Neil W and Keane, Andy J.** 2006, Proceedings of the Royal Society, pp. 2177-2204.

[18] *Surrogate-assisted evolutionary computation: Recent advances and.* **Jin, Yaochu.** 2011, Elsevier, pp. 61-70.

[19] *A Taxonomy of Global Optimisation Methods Based on Response Surfaces.* **Jones, Donald R.** 2001, Journal of Global Optimisation, pp. 345-383.

[20] **Lang, Chao-yi.** Kriging Interpolation. *Dept. of Computer Science Cornell University.* [Online] 2009. [Cited: 10 12 2014.] http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-94to95/clang/kriging.html.

[21] **Makhnin, Oleg.** Lecture 10: Introduction to Kriging. *New Mexico Tech.* [Online] 11 10 2011. [Cited: 10 12 2014.] http://infohost.nmt.edu/~olegm/586/HYD10.pdf.

[22] **LEAP CFD.** Tips & Tricks: Inflation Layer Meshing in ANSYS. *Leap Australia Computational Fluid Dynamics.* [Online] 06 01 2012. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/tips-tricks-inflation-layer-meshing-in-ansys/.

[23] **LEAP CFD.** Tips & Tricks: Estimating the First Cell Height for Correct Y+. *Leap Australia Compuational Fluid Dynamics.* [Online] 01 07 2013. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/tips-tricks-cfd-estimate-first-cell-height/.

[24] **LEAP CFD.** Turbulence Part 3 - Selection of wall functions and Y+ to best capture the Turbulent Boundary Layer. *Leap Australia Compuational Fluid Dynamics.* [Online] 12 04 2013. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/turbulence-part-3-selection-of-wall-functions-and-y-to-best-capture-the-turbulent-boundary-layer/.

[25] **LEAP CFD.** Turbulence Part 4 - Reviewing how well you have resolved the boundary layer. *Leap Australia Compuational Fluid Dynamics.* [Online] 06 05 2013. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/tips-tricks-turbulence-part-4-reviewing-how-well-you-have-resolved-the-boundary-layer/.

[26] **LEAP CFD.** Tips & Tricks: Turbulence Part 1 - Introduction to Turbulence Modelling. *Leap Australia Computational Fluid Dynamics.* [Online] 25 05 2012. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/turbulence-modelling/.

[27] **LEAP CFD.** Tips & Tricks: Convergence and Mesh Independence Study. *Leap Australia Computational Fluid Dynamics.* [Online] 17 01 201. [Cited: 01 04 2015.] http://www.computationalfluiddynamics.com.au/convergence-and-mesh-independent-study/.

[28] **Fasshauer, Greg.** Multistep Methods. *Illinois Institute of Technology.* [Online] 15 02 2007. [Cited: 02 04 2015.] http://www.math.iit.edu/~fass/478578_Chapter_2.pdf.

[29] **Blanco, Monica.** Fractional Fractorial Designs. *Universitat Rovira i Virgili.* [Online] 09 05 2006. [Cited: 03 04 2015.] http://www.etseq.urv.es/doctorat/index/running/2005_2007/courses_d/Experimental%20Design/ed%20fract%20factorial%20designs.pdf.

[30] **Beare, Dr Bob.** Numerical Optimisation. *Exeter Learning Environment - ECM2704.* [Online] 26 11 2014. [Cited: 05 04 2015.] http://vle.exeter.ac.uk/pluginfile.php/452559/mod_resource/content/5/lectures_20_to_25_ele.pdf.