Constraint Management of Reduced Representation Variables in Decomposition-Based Design Optimization

In decomposition-based design optimization strategies such as analytical target cascading (ATC), it is sometimes necessary to use reduced representations of highly discretized functional data exchanged among subproblems to enable efficient design optimization. However, the variables used by such reduced representation methods are often abstract, making it difficult to constrain them directly beyond simple bounds. This problem is usually addressed by implementing a penalty value-based heuristic that indirectly constrains the reduced representation variables. Although this approach is effective, it leads to many ATC iterations, which in turn yields an ill-conditioned optimization problem and an extensive runtime. To address these issues, this paper introduces a direct constraint management technique that augments the penalty value-based heuristic with constraints generated by support vector domain description (SVDD). A comparative ATC study between the existing and proposed constraint management methods involving electric vehicle design indicates that the SVDD augmentation is the most appropriate within decomposition-based design optimization. [DOI: 10.1115/1.4004976]

Keywords: decomposition-based design optimization, reduced representation, constraint management, support vector domain description

1 Introduction

Complex design problems are often addressed through a decomposition and collaboration process. In the development of an electric vehicle (EV) powertrain, for example, engineers may be interested in key components such as the battery, electric traction motors, and belt-drive transmission system. Since the design and integration of each component can be challenging to address simultaneously in an optimization framework, this problem may be split into two subproblems: a system-level problem that deals with the design of the battery and belt-drive transmission along with the selection of motor performance curves such that maximum energy efficiency is achieved (while balancing other performance requirements), and a subsystem-level problem that ensures that the motor designs meet the performance prescribed at the system level. Hence, although such division of labor may expedite the design process, collaboration is still required in order to provide a single, realizable design solution that satisfies all of the criteria in these decomposition-based optimization strategies.

One technique that captures this process effectively is analytical target cascading (ATC) [1,2]. In ATC, coupled quantities exchanged between subproblems are treated as decision variables. Sometimes these coupling variables may consist of highly discretized functional data from a "blackbox" simulation, such as the motor performance curves in the EV powertrain design problem. The discretized functional data can be nominally represented through a q-dimensional vector z that consists of prescribed, dependent functional data values. In turn, this vector can be conceptualized as part of an interpolation function for the continuous function

\[ z = f(y) \approx F([z_1, z_2, \ldots, z_q]^T, [y_1, y_2, \ldots, y_q]^T, y) \]  

where y denotes the independent variable, z denotes the dependent variable, y denotes the prescribed, independent functional data values, and F is an interpolation function or lookup table (as commonly seen in MATLAB®, for example). Because each element within z is a decision variable in ATC, the design problem can become prohibitively large for optimization. Therefore, it becomes necessary to use reduced representations of the functional data that improve optimization efficiency and maintain reasonable accuracy [3,4]. However, many times the variables used by reduced representations are abstract, thus leading to difficulties in constraining their decision space appropriately. Clearly, this can cause ill-behaved and/or failed analysis and optimization as the optimizer may select decision vectors outside of the feasible space. This constraint management issue is usually addressed by using a penalty value-based heuristic to indirectly constrain the reduced representation variables. While this approach is effective, it is not efficient; it often requires many ATC iterations, leading to an ill-conditioned optimization problem and an extensive runtime.

This work resolves the issues with the current constraint management method by introducing a new, more direct approach in which the penalty value-based heuristic is augmented with constraints generated by support vector domain description (SVDD). The paper is organized as follows: Section 2 provides some background on reduced representations as well as the issue of constraint management for abstract reduced representation variables, Sec. 3 presents a reduced representation method known as proper orthogonal decomposition (POD), Sec. 4 briefly reviews ATC, Sec. 5 discusses the two constraint management methods and how they are implemented in an optimization framework, Sec. 6 applies the constraint management methods in an EV powertrain design study, and Sec. 7 offers some conclusions.

2 Background

Reduced representations are broadly defined as techniques that minimize the dimensionality of the vector representation of highly
discretized functional data such that optimization efficiency is significantly improved while sufficient accuracy is preserved [3,4]. These methods include metamodels that use low-dimensional inputs [5] as well as curve-fitting models that use coefficients of basis functions [4] as reduced representation variables, respectively. In general, curve-fitting approaches are more appropriate for reduced representation since they are unlikely to use variables that violate the necessary condition of additive-separability [6] for decomposition-based optimization strategies. POD [7,8] is among the most attractive curve-fitting approaches as it uses data samples (rather than assumptions) to determine its basis functions, requires limited assumptions regarding the number of coefficients to use, and needs only a relatively small number of such coefficients for approximation [4].

Because these coefficients are not physically meaningful decision variables in an optimization framework, it is extremely challenging to constrain them beyond simple bound constraints. Of course, failure to properly constrain these variables may cause the optimizer to select decision vectors that are outside of the validity region (and hence decision space) of the reduced representation model, thus leading to errant and/or failed analysis and optimization as in Ref. [9]. A penalty value-based heuristic [4] is therefore typically used to constrain the reduced representation variables. This involves assigning large penalty values to objective and constraint function outputs that depend on the reduced representation variables when the optimizer selects a decision vector outside their model validity region (often observed when the related analysis functions/simulations fail). Such an approach effectively forces the selection of reduced representation decision vectors that are within the model validity region. However, this frequently occurs at the expense of additional ATC iterations, leading to an ill-conditioned optimization problem and an extensive runtime. It is therefore preferable to implement a constraint management technique that could resolve these issues.

Although methods such as probability-based density models [10], convex hulling algorithms [11], and support vector machines [12,13] are reasonable candidates for a new constraint management technique, SVDD [14,15] is preferable as it can generate boundary constraints for high-dimensional, nonconvex datasets consisting of a single class using a moderate number of samples. This is certainly the case with reduced representations using curve-fitting models similar to POD, where the number of representation variables may still be large on an absolute basis (i.e., not relative to the original vector representation of the functional data), their decision space is rarely convex, and only a single class of data containing a moderate number of samples is known. Furthermore, SVDD has been successfully implemented in very similar applications for predictive modeling [16,17] but with physically meaningful, observational input data variables. One caveat that still exists is that many optimizers periodically violate constraints during the solution process, which in this study could lead to the failure of underlying analysis/simulation models dependent on the reduced representation variables. Therefore, in this case, it is still useful and necessary to use the penalty value-based heuristic with SVDD (since SVDD alone cannot detect and circumvent model failures). It is expected, however, that the inclusion of the SVDD-related constraints would minimize this possibility of failure as well as improve the problem condition and runtime. This is because the optimizer would have to satisfy such constraints for all feasible decision vectors and hence spend more time (i.e., more function evaluations) within the feasible domain.

3 Proper Orthogonal Decomposition

POD [7,8] is a model reduction technique that is often used in the engineering applications to facilitate the analysis, the design, and the optimization of systems with extremely large data representations. In broader applications, POD is also referred to as Karhunen–Loeve expansion [18,19] or principal component analysis [20]. Mathematically, all of these terms refer to the same linear transformation method, but with a specific meaning in various fields. For systems with discrete data representations, POD reduces the original data representations according to

\[ \mathbf{z} = \mathbf{\Phi}_p \mathbf{x}_p + \mathbf{z} \]  

(2)

where \( \mathbf{z} \) is the original data representation of dimension \( q \), \( \mathbf{z}_p \) is the reduced data representation of dimension \( p \ll q \), and \( \mathbf{\Phi}_p \) is a matrix of the \( p \) most energetic basis vectors \( \mathbf{\phi} \) used to construct the approximation of the original data representation. The final term \( \mathbf{z} \) is the sample mean vector of dimension \( q \) and is used to center the data for the approximation. Note that in this study, \( \mathbf{z} \) consists of functional data, and so the basis vectors can be conceptualized as basis functions. These are in turn scaled by each element within \( \mathbf{z}_p \), which are referred to as POD coefficients. POD ultimately involves the construction of the \((q \times q)\) or \((q \times m)\) full basis matrix \( \mathbf{\Phi} \) (where its dimensionality depends on the solution method) based on \( m \) samples \( \mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{iq}]^T \) (where \( m \) is sufficiently large) and its reduction by examining each basis vector’s contribution toward representing the original sample set. This is accomplished by using either the direct method or the method of snapshots [7].

The most efficient approach when \( q \leq m \) is the direct method, which begins by forming the covariance matrix \( \mathbf{R} \)

\[ \mathbf{R} = \frac{\mathbf{Z} \mathbf{Z}^T}{m} \]  

(3)

In the above, \( \mathbf{Z} \) is a \((q \times m)\) matrix containing all the samples of the original data representation and \( \mathbf{Z} \) is a \((q \times m)\) matrix of the sample mean vector repeated \( m \) times. Next, a \((q \times q)\) eigenvalue problem on \( \mathbf{R} \) is used to construct \( \mathbf{\Phi} \)

\[ \mathbf{R} \mathbf{\Phi} = \mathbf{\Phi} \Lambda \]  

(4)

where \( \Lambda \) is the diagonal matrix of eigenvalues. Assuming that the basis vectors in \( \mathbf{\Phi} \) are arranged according to the magnitude of their associated eigenvalues

\[ \mathbf{\Phi} = [\mathbf{\phi}_1 \ \mathbf{\phi}_2 \ \cdots \ \mathbf{\phi}_q]^T, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \]  

(5)

this matrix is reduced to \( \mathbf{\Phi}_p \) based on the cumulative percentage variation (CPV). The CPV is a measure of the relative importance of each basis vector in \( \mathbf{\Phi} \)

\[ \sum_{i=1}^{p} \lambda_i \times 100 \geq \text{CPV}_{\text{goal}} \]  

(6)

Observe that \( \text{CPV}_{\text{goal}} \) is assigned based on the desired amount of information to be captured through POD, which is usually 99% or higher [22].

When \( q > m \), the most efficient solution technique is the method of snapshots [7]. This time, a correlation matrix \( \mathbf{R} \) is generated

\[ \mathbf{R} = \frac{\mathbf{Z} \mathbf{Z}^T}{m} \]  

(7)

From here, the associated \((m \times m)\) eigenvalue problem is solved

\[ \mathbf{RV} = \mathbf{VA} \]  

(8)

where \( \mathbf{V} \) represents the matrix of eigenvectors. The \((q \times m)\) orthogonal basis matrix is constructed from

\[ \mathbf{\Phi} = \mathbf{ZV}_p, \quad v_{pq} = \frac{1}{\sqrt{m \lambda_q}} v_{ij} \]  

(9)

The above equations demonstrate why this procedure is referred to as the method of snapshots: each basis vector is a linear combination of the \( m \) sample vectors, or “snapshots,” of original data [7]. Finally, \( \mathbf{\Phi}_p \) is determined using the same procedure outlined in Eqs. (5) and (6) with \( q \) replaced by \( m \).
4 Analytical Target Cascading

ATC [1,2] is a decomposition-based optimization strategy that simultaneously minimizes performance-related objectives and deviations between design targets cascaded from upper levels and their realizable responses at lower levels. Optimality is achieved when the targets and responses are within an acceptable tolerance of one another.

The strategy begins by decomposing the system into design subproblems, where the top level is referred to as the system level and lower levels are referred to as subsystem levels. Note that a subproblem linked above a given element of interest is called a parent, and subproblems linked below a given element of interest are called children. The general ATC subproblem $P_i$ for the $i$th level and the $j$th element is defined as [23]

$$
\begin{align*}
\min_{x_i} & \quad f_i(x_i) + \pi(c(x_{11}, \ldots, x_{NM})) \\
\text{subject to} & \quad g_i(x_i) \leq 0, \quad h_i(x_i) = 0 \\
\text{where} & \quad x_i = [x_i, r_j, t_{(i+1)k_1}, \ldots, t_{(i+1)k_\alpha}], \\
& \quad c = [c_1, \ldots, c_{NM}]
\end{align*}
$$

(10)

Note that the linear and quadratic terms in the AL penalty function are weighted by the vectors $v$ and $w$, respectively. These decomposed problems are solved in an inner loop strategy where the weights remain constant. After inner loop convergence, termination conditions are evaluated in the outer loop, and if another inner loop execution is required the penalty weights are updated according to the following scheme:

$$
\begin{align*}
v^{(K+1)} &= v^{(K)} + 2w^{(K)} \circ w^{(K)} \circ c^{(K)} \\
w^{(K+1)} &= \beta w^{(K)}, \quad \text{where } \beta \geq 1
\end{align*}
$$

(12)

The information flow for the general ATC-AL subproblem is illustrated in Fig. 1.

5 Constraint Management of Reduced Representation Variables

As mentioned in Sec. 2, the penalty value-based heuristic for constraining abstract reduced representation variables yielded reasonably accurate design solutions but was not efficient. Indeed, a similar EV powertrain design study using this approach required 83 ATC iterations, leading to an ill-conditioned ATC problem with excessive large AL penalty function weights and a much longer runtime (59 h) than what is commonly accepted in practice [4]. It is therefore preferable to consider an alternative constraint management approach in which the penalty value-based heuristic is augmented by explicit constraints generated by SVDD. This section presents the details of both constraint management methods for the abstract reduced representation variables.

5.1 Penalty Value-Based Heuristic. The penalty value-based heuristic assigns large penalty values to objective function and constraint function outputs that depend on reduced representation variables when the optimizer selects a decision vector that is outside their model validity region. Such a condition is usually indicated when the associated analysis/simulation models fail or produced significantly errant values. Theoretically, the penalty values would indirectly force the selection of reduced representation variables that lie within the model validity region. A key assumption for the successful implementation of this method is that a nongradient-based optimizer will be used instead of a gradient-based optimizer. This is because penalizing outputs such as the objective function with large values in gradient-based optimizers can result in ill-conditioned problems due to large gradients.

When programming in MATLAB®, a reasonable way to implement this technique would be to use a “try-catch” statement [24]. For example, MATLAB® could attempt to run the analysis/simulation models that are dependent on the reduced representation variables between the keywords “try” and “catch” and return the results if the models ran successfully. However, if this is not the case, then MATLAB® could penalize the relevant outputs between the keywords “catch” and “end”.

5.2 Support Vector Domain Description. SVDD [14,15] is a classification method that uses a machine learning algorithm to approximate the boundary of a set of data points and to identify whether new data points lie inside the boundary description. In...
particular, SVDD can be used to represent data set boundaries that are nonlinear, nonconvex, and even disconnected without adding much complexity or computational burden. It is also distinct from other machine learning algorithms in that it requires only one class of data for classification since it aims to identify the minimum radius hypersphere enclosing the data. This feature is advantageous for classification problems in which a second class of data is either unknown or difficult to generate, as is the case for the reduced representation variables.

SVDD begins with assumption that the data space (or reduced representation model validity region, in our case) can be effectively characterized by a hypersphere [14,15]. Since the associated primal optimization problem (see Appendix A) for SVDD is never solved for reasons given in Ref. [25], the dual optimization problem formulation is used

$$\begin{align*}
\max_{B_i} & \sum_i B_i(z_i^T z_i) - \sum_{i,j} B_i B_j(z_i^T z_j) \\
\text{subject to} & \ 0 \leq B_i \leq C_p, \quad i = 1, \ldots, m \\
& \sum_i B_i = 1
\end{align*}$$

where $B_i$ denotes the dual variable, $C_p$ denotes the slack variable penalty constant (from the primal formulation), $z_i$ denotes a data sample (which is a $p$-dimensional vector of reduced representation variables in this application), and $m$ denotes the number of samples. The solutions are categorized according to three conditions: $B_i = 0$, $0 < B_i < C_p$, and $B_i = C_p$. The first condition ($B_i = 0$) is satisfied by the majority of the dual variables for sufficiently large $m$ [16] and implies that the associated sample $z_i$ lies within the hypersphere. The second condition ($0 < B_i < C_p$) implies that the associated sample $z_i$ lies at the boundary of hypersphere and is essential to its description; these samples are termed support vectors [14–16]. The third condition ($B_i = C_p$) implies that the associated sample $z_i$ lies outside the hypersphere and is an outlier.

Using the dual variables along with the following constraint on the hypersphere center $a$

$$a = \frac{\sum_i B_i z_i}{\sum_i B_i}$$

the squared distance $R_{\text{hyp}}^2$ from a to any arbitrary data point $z_{a}$ is calculated as

$$R_{\text{hyp}}^2(z_{a}) = \|z_{a} - a\|^2 = z_{a}^T z_{a} - 2 \sum_i B_i(z_i^T z_{a}) + \sum_{i,j} B_i B_j(z_i^T z_j)$$

where the indices $i$ and $j$ run over the support vectors and their associated dual variables. With this definition, $R_{\text{hyp}}$ can be calculated by setting $z_{a} = z_{j}$ for any sample that is a support vector, and in turn this information can be used to determine whether an arbitrary data point lies inside the boundary description

$$R_{\text{dist}}^2(z_{a}) \leq R_{\text{hyp}}^2$$

Such a condition can be added to a design optimization problem to constrain the abstract reduced representation variables directly. Although SVDD assumes a hyperspherical data space, it can still be used in the more likely situation of nonhyperspherical data spaces. This simply requires the data to be mapped into some higher-dimensional “feature space” through a nonlinear transformation where the hyperspherical domain assumption is more appropriate [16,17]. Because these transformations can be difficult to develop explicitly, Mercer kernel functions [26] are used to represent the dot product between any two nonlinear transformations. The most preferred in the literature is the Gaussian kernel function

$$K_G(z_{a}, z_{j}) = \exp(-\theta \|z_{a} - z_{j}\|^2)$$

where $\theta$ is the kernel width parameter. Equation (17) can then be substituted for the dot product terms in Eqs. (13) and (15), yielding the following dual optimization problem and squared distance formulations that are used in most applications

$$\begin{align*}
\max_{B_i} & \sum_i B_i K_G(z_i, z_i) - \sum_{i,j} B_i B_j K_G(z_i, z_j) \\
\text{subject to} & \ 0 \leq B_i \leq C_p, \quad i = 1, \ldots, m \\
& \sum_i B_i = 1 \\
R_{\text{dist}}^2(z_{a}) &= K_G(z_{a}, z_{a}) - 2 \sum_i B_i K_G(z_{a}, z_i) + \sum_{i,j} B_i B_j K_G(z_i, z_j)
\end{align*}$$

The parameters $\theta$ and $C_p$ in Eqs. (18) and (19) must be tuned to construct an appropriate SVDD. In practice, however, modifications to $C_p$ have a minimal impact on the solution [14–16], leaving only $\theta$ to be tuned. This parameter is adjusted using the leave-one-out method [25] such that overfitting of the data is minimized. Specifically, this tuning method states that the probability of overfitting can be estimated by determining the proportion of samples that are support vectors [14,15]

$$E[P(\text{error})] = \frac{n_{SV}}{m}$$

where $n_{SV}$ refers to the number of support vectors. Hence, $\theta$ can be determined by setting an acceptable overfitting target $P_{\text{target}}$ and minimizing the error between this target and the estimated SVDD performance indicated by Eq. (20). Note that underfitting error cannot be addressed as this requires samples outside the target domain and hence violates the assumption of a single data class for SVDD. For a brief illustrative example of SVDD, refer to Appendix A.

6 Electric Vehicle Powertrain Optimization

The design application, in which the constraint management techniques were assessed, was for an EV powertrain system. Details for this model, which was developed in a MATLAB®/SIMULINK® environment, can be seen in Refs. [3, 27]. Figure 2 shows the vehicle configuration, which is for a two-passenger, minicompact vehicle designed primarily for urban driving with some highway speed capability. This classification is evident by the vehicle’s overall dimensions, which includes a wheelbase of $L = 1.80$ m and a track width of $W = 1.27$ m. The
vehicle is powered by a lithium-ion battery energy storage system, which can vary in length, width, and longitudinal location relative to the front end of the battery compartment such that it lies within the dashed region defined by \( b_{\text{max}} = 1.05 \) m and width \( b_{\text{max}} = 1.20 \) m. Two electric traction motors drive the rear wheels through a synchronous belt-drive system and are mounted at the pivots on the rear suspension trailing arms to reduce the unsprung mass in the system. A MacPherson strut configuration is used for the front suspension, and finally, low rolling resistance P145/70R12 tires are used to minimize the energy consumption. All physically meaningful decision variables (i.e., non-reduced representation variables) for the ATC design problem formulation are listed in Table 1.

### 6.1 Reduced Representations: POD

Since the ATC design problem formulation required the highly discretized motor maps to become decision variables during optimization, reduced representation was necessary. Three POD models were developed to approximate functional data vectors associated with the maximum and minimum motor torque curves and the power loss map

\[
\begin{align*}
\Phi_{\max} & \approx \Phi_{\max}^T r_{\text{max}} + z_{\text{max}} \\
\Phi_{\min} & \approx \Phi_{\min}^T r_{\text{min}} + z_{\text{min}} \\
\Phi_{\text{p-Loss}} & \approx \Phi_{\text{p-Loss}}^T r_{\text{p-Loss}} + z_{\text{p-Loss}}
\end{align*}
\]

The functional data vectors \( \Phi_{\max} \) and \( \Phi_{\min} \) contained \( q_{\text{max}} = q_{\text{min}} = 41 \) discretized points each, whereas the functional data vector \( \Phi_{\text{p-Loss}} \) contained \( q_{\text{p-Loss}} = 3321 \) discretized points. The sample functional data vectors used to construct the POD representations were generated from an electric traction motor analysis model [3, 27] through a Latin hypercube experimental design of \( m = 500 \) motor maps (see Appendix B). Because \( q_{\text{max}} = q_{\text{min}} \ll m \), the direct method was used to develop \( \Phi_{\max} \) and \( \Phi_{\min} \), whereas the method of snapshots was used to develop \( \Phi_{\text{p-Loss}} \) since \( q_{\text{p-Loss}} \gg m \). The CPV was set to CPV goal = 99.99% based on part in the literature [22] as well as previous work [28]. This resulted in reduced representations \( \Phi_{\max} \), \( \Phi_{\min} \), and \( \Phi_{\text{p-Loss}} \) of dimension \( p_{\max} = 14 \), \( p_{\min} = 13 \), and \( p_{\text{p-Loss}} = 89 \), respectively. Hence, the combined dimensionality of the functional data vectors was reduced from \( Q = q_{\text{max}} + q_{\text{min}} + q_{\text{p-Loss}} = 3403 \) to \( Q = p_{\max} + p_{\min} + p_{\text{p-Loss}} = 116 \).

### 6.2 ATC Problem Formulation

The ATC problem formulation for the EV powertrain model consists of a two-level hierarchical decomposition. Battery and belt-drive transmission design as well as motor map selection is performed at the top-level subproblem, whereas detailed motor design is performed at the bottom-level subproblem. The top-level objective is to minimize the gasoline-equivalent fuel economy mpg, while minimizing inconsistencies with the bottom-level subproblem (through \( n \)), while the bottom-level objective is to minimize the inconsistency with the top-level subproblem. Although both subproblems are subject to decision variable bound constraints, only the top-level contains additional constraints based on battery packaging, acceleration performance, motor feasibility, vehicle range, power availability, and battery capacity.

Applying Eq. (11) directly, the vehicle subproblem \( P_{11} \), excluding decision variable bound constraints, is formulated as

\[
\min_{\mathbf{x}} -mp_g(\mathbf{x}_{11}) + \sum_{i} t_{22} + ||w_{22} \circ (t_{22} - r_{22})||^2_F
\]

subject to

\[
\begin{align*}
g_{11,1} & = b_{11} (\mathbf{x}_{11}) \leq 0 \\
g_{11,2} & = b_{11} (\mathbf{x}_{11}) \leq 0 \\
g_{11,3} & = b_{11} (\mathbf{x}_{11}) - b_{11,\text{max}} \leq 0 \\
g_{11,4} & = b_{11} (\mathbf{x}_{11}) \leq 0 \\
g_{11,8} & = c_{11} (\mathbf{x}_{11}) - c_{11,\text{max}} (\mathbf{x}_{11}) \leq 0
\end{align*}
\]

where \( g_{11,1} \) and \( g_{11,2} \) are battery width and length packaging constraints, \( g_{11,3} \) is a performance (0-60 mph acceleration time) constraint, \( g_{11,4} \) and \( g_{11,5} \) are motor torque and speed feasibility constraints, \( g_{11,6} \) is a vehicle range constraint, \( g_{11,7} \) is a power availability constraint, and \( g_{11,8} \) is a battery capacity constraint. The vectors \( z_{\text{comb}} = [z_{\text{max}}, z_{\text{min}}, z_{\text{p-Loss}}] \) and \( z_{\text{p-Comb}} = [z_{\text{max}}, z_{\text{min}}, z_{\text{p-Comb}}] \) refer to the combined vector of functional data variables and the combined vector of reduced representation variables, respectively. Additionally, the vectors \( t_{22} \) and \( r_{22} \) include six scalar-valued coupling variables: \( o_{\text{max}}, m_{\text{F}}, f_{\text{r}}, f_{\text{y}}, f_{\text{t}} \), and \( y_{\text{m}} \). Finally, the superscripts \( T \) and \( R \) denote target and response versions of the same coupling variable, respectively. The motor subproblem \( P_{22} \), excluding decision variable bound constraints, is formulated in a similar manner as

\[
\min_{\mathbf{r}_{22}} -v_{22}^T \mathbf{r}_{22} + ||w_{22} \circ (t_{22} - r_{22})||^2_F
\]

where

\[
\begin{align*}
t_{22} & = [t_{\text{max}}, t_{\text{min}}, n_e, R_e] \\
r_{22} & = [r_{\text{comb}}, o_{\text{max}}, m_{\text{R}}, f_{\text{r}}, f_{\text{y}}, f_{\text{t}}, y_{\text{m}}]
\end{align*}
\]

The problem formulation shown in Eqs. (24) and (25) was solved using the penalty value-based heuristic and its SVDD-alternative as constraint management methods for the reduced representation variables in the \( P_{11} \) subproblem. NOMADm [29] is a derivative-free optimization software package based on mesh-adaptive search algorithms, was used as the optimizer. The default settings were modified for the \( P_{11} \) subproblem such that only a Latin hypercube search was performed and 1000 function evaluations were permitted. This was necessary to alleviate computational issues associated with memory availability. However, for the \( P_{22} \) subproblem, the default settings were sufficient. Finally, in the ATC coordination strategy, the weight update parameter was set to \( \beta = 2.75 \), the initial weight vectors for both subproblems were set to \( v = 0 \) and \( w = 1 \), and the tolerance on \( \|e^T e - c^T c\|_\infty \) for outer loop convergence was set to \( 10^{-6} \). All computational work was performed on a 3 GHz, 4 MB RAM, Intel® Core™ 2 Duo CPU.

### 6.3 Constraint Management via Penalty Value-Based Heuristic

In order to implement the penalty value-based heuristic for constraint management, a MATLAB® try-catch conditional
Fig. 3 Penalty value-based heuristic: MATLAB try-catch statement

statement was written as seen in Fig. 3. This attempts to perform the powertrain simulations and, upon failing, returns infinite values as appropriate for $mpge$, $t_{hyp}$, $R$, and $P_f$. Note that $b_{n,1}$, $b_{r,1}$, and $C_0$ are not penalized since they are independent of the reduced representation variables, while $t_i$ and $o_i$ are not included because they inherently penalize inaccurate motor maps when the reduced representation variables are outside their model validity region.

Tables 2–4 show the ATC optimization results when implementing this constraint management technique. Convergence was achieved after 12 ATC iterations with a runtime of approximately 10.72 hours and resulted in a system solution that was reasonably consistent between both subproblems. The only active constraints were the upper bound on $o_{max}$, the performance constraint $g_{11.3}$, and the battery capacity constraint $g_{11.3}$ in the $P_{11}$ subproblem; these were limited to $o_{max} = 755$ rad/s, $t_{max} = 10$ s, and $C_{max} = 200$ Ah, respectively. The optimal values of the POD coefficients are not listed here as they are too numerous and not physically meaningful; however, the optimal motor map computed by these reduced representation variables is shown in Fig. 4.

Finally, the total mass of the vehicle was 1111 kg, with approximately 14.3% (158 kg) of the mass associated with the battery. These design conditions indicated that the EV could achieve a gasoline-equivalent fuel economy of $mpge = 184$ mpg and a range of $R = 134$ miles.

6.4 Constraint Management via SVDD Augmentation. In order to implement the SVDD augmentation for constraint management, three kernel-based SVDD models were developed using Eqs. (17)–(19) along with the tuning requirement given in Eq. (20) to approximate the POD model validity regions for $z_{r,\max}$, $z_{r,\min}$, and $z_{p,loss}$. This generated the following additional constraints for the $P_{11}$ subproblem in ATC:

$$g_{11.9} = \frac{R_{dist,max}(s_{11}) - R_{hyp,max}}{C_0} \leq 0$$
$$g_{11.10} = \frac{R_{dist,min}(s_{11}) - R_{hyp,min}}{C_0} \leq 0$$
$$g_{11.11} = \frac{R_{dist,loss}(s_{11}) - R_{hyp,loss}}{C_0} \leq 0$$

Figures 5–7 display portions of the optimal SVDD boundaries for the first two dimensions of the POD model validity regions for $z_{r,\max}$, $z_{r,\min}$, and $z_{p,loss}$. Although the boundaries may appear “loose” (or in the case of the $z_{p,loss}$ nonexistent), it is noted that the data set are multidimensional and hence what appears “loose” in one 2D projection may be “tight” in another 2D projection. The samples used to construct the SVDD models were identical to those used for the POD representations but mapped appropriately.

Fig. 4 Optimal motor map, PVBH

Fig. 5 Partial SVDD boundary, max-torque POD model validity region

Table 2 Optimal decision vector for $P_{11}$ subproblem, PVBH

<table>
<thead>
<tr>
<th>$B_t$</th>
<th>$B_W$</th>
<th>$B_L$</th>
<th>$v_{batt}$</th>
<th>$P_r$</th>
<th>$c_{max}$</th>
<th>$m_r$</th>
<th>$r$</th>
<th>$I_{pm}$</th>
<th>$I_{im}$</th>
<th>$y_{im}$</th>
<th>$y_{r}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.74</td>
<td>1.43</td>
<td>19.75</td>
<td>0.25</td>
<td>3.13</td>
<td>755</td>
<td>40.39</td>
<td>0.28</td>
<td>1.12</td>
<td>1.20</td>
<td>0.39</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Optimal decision vector for $P_{22}$ subproblem, PVBH

<table>
<thead>
<tr>
<th>$l_s$</th>
<th>$r_m$</th>
<th>$r_n$</th>
<th>$R_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.098</td>
<td>0.123</td>
<td>17.62</td>
<td>0.053</td>
</tr>
</tbody>
</table>

Table 4 Optimal consistency constraint vector/weights, PVBH

<table>
<thead>
<tr>
<th>Consistency constraint</th>
<th>$c_{opt}$</th>
<th>$v_{opt}$</th>
<th>$w_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{r,\max}$</td>
<td>0.45</td>
<td>6.37e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{r,\min}$</td>
<td>0.41</td>
<td>5.82e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{p,loss}$</td>
<td>0.73</td>
<td>1.01e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,\max}$</td>
<td>0</td>
<td>0</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,\min}$</td>
<td>-0.46</td>
<td>-6.53e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,fr}$</td>
<td>0</td>
<td>1.51e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,\ym}$</td>
<td>0</td>
<td>5.93e+06</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,\om}$</td>
<td>-0.02</td>
<td>-3.21e+07</td>
<td>6.80e+04</td>
</tr>
<tr>
<td>$c_{t,\om}$</td>
<td>0</td>
<td>3.67e+06</td>
<td>6.80e+04</td>
</tr>
</tbody>
</table>
SVDD constraints, thus likely leading to model failure which in this study since the optimizer may periodically violate the conditions, one could reasonably conclude it would offer greater benefits when used more suitably.

The ATC optimization results when implementing the SVDD augmentation as a constraint management technique for the reduced representation variables. This could in turn provide suboptimal results in terms of accuracy and even efficiency when compared to the exclusive penalty value-based heuristic; therefore, if the SVDD augmentation could succeed under such unfavorable conditions, one could reasonably conclude it would offer greater benefits when used more suitably.

Table 5 Optimal decision vector for $P_{11}$ subproblem, SVDD augmentation

<table>
<thead>
<tr>
<th>$B_f$</th>
<th>$B_w$</th>
<th>$B_L$</th>
<th>$n_{min}$</th>
<th>$p_r$</th>
<th>$\omega_{max}^{f}$</th>
<th>$m_{m}^{f}$</th>
<th>$J_r^{f}$</th>
<th>$I_{pr}^{f}$</th>
<th>$I_{pm}^{f}$</th>
<th>$y_m^{f}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.74</td>
<td>1.43</td>
<td>19.75</td>
<td>0.25</td>
<td>3.93</td>
<td>755</td>
<td>40.39</td>
<td>0.28</td>
<td>1.12</td>
<td>1.20</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 6 Optimal decision vector for $P_{22}$ subproblem, SVDD augmentation

<table>
<thead>
<tr>
<th>$I_r$</th>
<th>$r_m$</th>
<th>$n_r$</th>
<th>$R_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.096</td>
<td>0.124</td>
<td>17.87</td>
<td>0.065</td>
</tr>
</tbody>
</table>

Table 7 Optimal consistency constraint vector/weights, SVDD augmentation

<table>
<thead>
<tr>
<th>Consistency constraint</th>
<th>$c_{opt}$</th>
<th>$v_{opt}$</th>
<th>$w_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_{f,\text{max}}$</td>
<td>0.45</td>
<td>449</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f,\text{min}}$</td>
<td>0.42</td>
<td>416</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f,\text{pl,less}}$</td>
<td>0.29</td>
<td>297</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{\text{con}}$</td>
<td>0</td>
<td>0</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f}$</td>
<td>0</td>
<td>0.019</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f,\text{con}}$</td>
<td>0</td>
<td>0</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f,\text{con}}$</td>
<td>0</td>
<td>0.035</td>
<td>57.2</td>
</tr>
<tr>
<td>$c_{f,\text{con}}$</td>
<td>0</td>
<td>0.025</td>
<td>57.2</td>
</tr>
</tbody>
</table>

6.5 Summary of ATC Results. It is clear from the results that the SVDD augmentation improves the efficiency of ATC compared with the penalty value-based heuristic. Indeed, the runtimes associated with the penalty value-based heuristic and the SVDD augmentation were 10.72 h and 3.95 h, respectively. The improved efficiency using the SVDD augmentation was due to the explicit constraints imposed on the POD model validity regions, which enabled the optimizer spent less time (i.e., fewer function evaluations) exploring designs outside the feasible decision space. Nevertheless, although the SVDD augmentation reduced the computational time during optimization, it still required significant modeling time offline. Therefore, it is more appropriate to consider the total computational effort (modeling time plus runtime) when assessing any efficiency gains via SVDD. The modeling times required to construct the optimal SVDDs for the POD model validity regions of $x_{e,\text{max}}$, $x_{e,\text{min}}$ and $x_{e,\text{pl,less}}$ were 0.94 h, 1.13 h, and 0.27 h, respectively. Because the total computational effort (6.29 h) associated with the SVDD augmentation was less than the runtime associated with the penalty value-based heuristic, it is clear that the SVDD augmentation was more computationally efficient.
achieved our main objective of producing better-conditioned optimization problems with significantly reduced runtimes. While it is expected that SVDD’s computational savings would be observed for any optimization problem (including AiO problems) containing abstract decision variables, the most significant payoff is for decomposition-based optimization problems since these strategies iteratively solve subproblems within their structure. Additionally, the SVDD augmentation can yield more accurate design solutions, which is compelling when one considers that a “worst case” comparative study was performed by setting a high overfitting target for the SVDD models. In particular, we can anticipate even more accuracy improvements when the overfitting target is set more appropriately. This is of course a challenge since setting this parameter too conservatively yields poor boundary descriptions, while setting it too aggressively can possibly truncate a design region containing the optimal solution. The systematic balance of these tradeoffs is proposed as a topic of future work.

Acknowledgment

This research has been partially supported by the Automotive Research Center, a U.S. Army RDECOM Center of Excellence headquartered at the University of Michigan. This support is gratefully acknowledged.

Appendix A

Formally, the objective of SVDD is to solve the following primal optimization problem:

$$\min_{R_{hyp}, \alpha, \xi_i} R_{hyp}^2 + C_p \sum \xi_i$$

subject to $$\| z_{ij} - \alpha \|_2^2 \leq R_{hyp}^2 + \xi_i, \quad i = 1, \ldots, m$$

where $$R_{hyp}$$ denotes the hypersphere radius, $$\xi_i$$ denotes a hypersphere radius slack variable, $$C_p$$ denotes the slack variable penalty constant, $$z_{ij}$$ denotes a data sample, $$\alpha$$ denotes the hypersphere center, and $$m$$ denotes the number of samples. The second term in the objective function of Eq. (A1) relaxes the optimization problem and permits the inclusion of outliers. To switch to the dual optimization problem, we must first construct the Lagrangian

$$L(R_{hyp}, \alpha, B_i, \xi_i, \mu_i) = R_{hyp}^2 + C_p \sum \xi_i$$

$$- \sum B_i \left( R_{hyp}^2 + \xi_i - \| z_{ij} - \alpha \|_2^2 \right)$$

with non-negative Lagrange multipliers $$B_i$$ and $$\mu_i$$. Applying Karush-Kuhn-Tucker conditions to Eq. (A2) yields the following constraints (along with the dual problem formulation):

$$\sum_i B_i = 1$$

$$\alpha = \sum_i B_i z_{ij}$$

$$C_p - B_i - \mu_i = 0, \quad i = 1, \ldots, m$$

The performance of SVDD is illustrated for the closed-curve shape in Fig. 9 that is bounded by the following functions:

$$f_1(x) = 10 \sin(\pi x) / \pi x, \quad x \in [-2.93, 2.93]$$

$$f_2(x) = \frac{1}{4} (x^4 - 5x^2 - 30), \quad x \in [-2.93, 2.93]$$
To generate an approximation of this boundary, SVDD was applied using $m = 300$ data samples and setting $C_p = 0.5$ and $P_{\text{target}} = 0.10$ appropriately based on experience. Figure 10 shows the optimal boundary approximation for the closed-curve shape. Although the SVDD approximation is not exact, it is reasonably accurate for the constraint management application that is described in this paper. That is, for a single class of known data (i.e., samples) and in the absence of physically meaningful supporting information, SVDD can generate an appropriate estimate of a data space boundary.

Appendix B

An error metric known as accuracy and validity algorithm for simulation (AVASIM) [30] was used to determine both the number of samples as well as the accuracy of the POD representations. This method characterizes the local and global error between original functional data and their approximations through $l_1$-norms and residual sums. Using these measures, error indices are constructed such that non-negative values of the combined index indicate valid approximations with accuracy levels between 0 and 1, and negative values of the combined error index generally indicate invalid approximations. Validity is defined by approximations that lie within some threshold value; therefore, a value of 0 indicates that the approximation is at the threshold and valid, whereas a value of 1 indicates that an approximation is completely accurate. Recently, this method was extended to two-dimensional functional data [31], which were necessary for the accuracy assessment of the power loss map in this study. An appropriate number of samples for the POD representations was then determined by ensuring that the combined error indices for the motor torque curves and power loss map were positive when averaged across all current samples and at least three standard deviations away from 0 (as set by the threshold value). This convention would theoretically ensure that the majority of the POD approximations of the motor maps would be reasonably accurate and valid. For the current study, this led to $m = 500$ samples when AVASIM was applied with a 10% tolerance (i.e., threshold value). As a point of further verification, POD approximations for the optimal motor map produced from the AiO optimization problem equivalent to the ATC optimization problem were measured through AVASIM with the same threshold value and found to be both reasonably accurate and valid (see Table 8 and Figs. 11 and 12).

![Fig. 9 Closed-curve shape for SVDD comparison](image)

![Fig. 10 SVDD boundary approximation for closed-curve shape](image)

![Fig. 11 Torque curve comparison for AiO optimal motor map](image)

![Fig. 12 Power loss map relative error for AiO optimal motor map](image)

Table 8 AVASIM results for POD-approximated AiO optimal motor map

<table>
<thead>
<tr>
<th>Index</th>
<th>Max-torque</th>
<th>Min-torque</th>
<th>Power loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{combined}}$</td>
<td>0.964</td>
<td>0.979</td>
<td>0.809</td>
</tr>
</tbody>
</table>
References