Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Quaternionic and complex-valued Support Vector Regression for Equalization and Function Approximation

Alistair Shilton and Daniel T. H. Lai

Abstract— **Support Vector Regressors (SVRs) are a class of nonlinear regressor inspired by Vapnik's Support Vector (SV) method for pattern classification. The standard SVR has been successfully applied to real number regression problems such as financial prediction and weather forecasting. However in some applications the domain of the function to be estimated may be more naturally and efficiently expressed using complex numbers (eg. communications channels) or quaternions (eg. 3-dimensional geometrical problems). Since SVRs have previously been proven to be efficient and accurate regressors, the extension of this method to complex numbers and quaternions is of great interest. In the present paper the standard SVR method is extended to cover regression in complex numbers and quaternions. Our method differs from existing approaches in-so-far as the cost function applied in the output space is rotationally invariant, which is important as in most cases it is the magnitude of the error in the output which is important, not the angle. We demonstrate the practical usefulness of this new formulation by considering the problem of communications channel equalization.**

I. Introduction

The problem of function estimation can be solved using supervised learning where the unknown function is assumed to model some generating system such as a chemical plant. This is commonly treated as a regression problem, where given a set of training data consisting of inputs and corresponding system outputs, the unknown generating function is estimated by regressing on the available data.

Support Vector regressors (SVRs) [1], [2], [3] are a class of nonlinear regressor inspired by Vapnik's SV formulation for classification [4], [5]. Like Vapnik's support vector classifiers, SVRs achieve nonlinearity by first implicitly mapping all data into a (usually) higher dimensional feature space. In this feature space, a linear function is constructed by minimizing a regularized cost function. The regularization term in the cost function is present to bias the estimated function toward functions with smaller gradient in feature space [6], [7] and hence minimise noise sensitivity.

The standard SVR formulation assumes an unknown generating function with real outputs. However, in many other areas such as telecommunications and geometrical problems, the target outputs are either complex numbers (eg. communication channels, 2-dimensional geometric problems) or quaternions (eg. 3-dimensional geometric problems), where quaternions are an extension of complex numbers and have 4 axes which represent 1 real and 3 imaginary components.

Alistair Shilton and Daniel T.H. Lai are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, Australia (email: apsh@ee.unimelb.edu.au and d.lai@ee.unimelb.edu.au, respectively).

While it is possible to use either 2 or 4 independent SVRs to separately treat each component (axis) of the complex number/quaternion output, this may not be desirable for a number of reasons. Firstly, the complex/quaternion system outputs may be coupled and treating them independently may degrade regressor performance. Secondly, as each of the independent SVRs are constructed using minimization of a risk function along a single axis, the overall risk function will not be rotationally symmetric. This means that changes in angles between the axes will affect the estimated function when only the magnitude of the error is considered to be of primal importance. Some previous work has been done toward the extension of the SVR framework to cover complex-valued and quaternion-valued regression [8], [9]. However, while this work has addressed the first point (ie. the problem of interconnectedness) it has not addressed the second. In this paper we demonstrate a new way of extending the SVR framework so that both points are addressed.

We begin by first reviewing the standard ϵ-SVR method in section II. This is important to set the background for our proposed extension to complex/quaternionic systems in section III. We start with the primal form and systematically work through the derivation to obtain the dual formulation. While the derivation of the dual is more complex than the standard ϵ-SVR case, the dual itself is directly analogous to the standard ϵ-SVR dual which means that it can be solved using ideas from existing SVR training algorithms. To demonstrate the practical applicability of our regressor, we apply it to a 4-QAM (Quadrature Amplitude Modulation) channel equalization problem [10] which requires a highly non-linear decision boundary.

A. Notation

Throughout this paper the quaternionic division algebra [11], [12], [13] will be denoted $\mathbb{H}$, the field of complex numbers $\mathbb{C}$, the completely ordered field of reals $\mathbb{R}$, the positive reals $\mathbb{R}^+$ and the negative reals $\mathbb{R}^-$. We use $\mathbb{N}$ to denote the natural numbers (including zero), $\mathbb{Z}$ the integers, and $\mathbb{Z}_n$ to represent the integers modulo $n \in \mathbb{N}$ (so $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$). For generality we let $\mathbb{X} \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$. An n-sphere of radius ϵ will be denoted $S^n_\epsilon(\mathbb{X})$, ie.:

$$S^n_\epsilon(\mathbb{X}) = \left\{ z \in \mathbb{X}^{n+1} \middle| \, |z| = \epsilon \right\}$$

For any $x \in \mathbb{H}$, $x_R \in \mathbb{R}$ is defined to be the real part of x and $x_I, x_J, x_K \in \mathbb{R}$ the imaginary parts of x, so in componentwise notation the quaternion x may be written $x = x_R + \mathbf{i}x_I + \mathbf{j}x_J + \mathbf{k}x_K$ using the standard $\mathbf{i}, \mathbf{j}, \mathbf{k}$ notation.

The conjugate and norm of x will be denoted $\bar{x}$ and $|x|$ respectively. As per [11], for $x \in \mathbb{H}$:

$$\begin{aligned}
&\bar{x} = x_R - \mathbf{i}x_I - \mathbf{j}x_J - \mathbf{k}x_K \\
&|x| = \sqrt{x\bar{x}} = \sqrt{x_R^2 + x_I^2 + x_J^2 + x_K^2} \in \mathbb{R}\backslash\mathbb{R}^- \\
&\mathrm{Re}\,(x) = \tfrac{1}{2}(x + \bar{x}) = x_R \in \mathbb{R} \\
&\mathrm{Pu}\,(x) = \tfrac{1}{2}(x - \bar{x}) = \mathbf{i}x_I + \mathbf{j}x_J + \mathbf{k}x_K \in \mathbb{H}\backslash\left(\mathbb{R}^+ \cup \mathbb{R}^-\right) \\
&\mathrm{Un}\,(x) = \tfrac{x}{|x|} \in S_1^0(\mathbb{H}, 0)\ \forall\, x \neq 0 \\
&\mathrm{sgn}\,(x) = \mathrm{sgn}\,(x_R) + \mathbf{i}\mathrm{sgn}\,(x_I) + \mathbf{j}\mathrm{sgn}\,(x_J) + \mathbf{k}\mathrm{sgn}\,(x_K) \\
&\nabla_x = \nabla_{x_R} + \mathbf{i}\nabla_{x_I} + \mathbf{j}\nabla_{x_J} + \mathbf{k}\nabla_{x_K}
\end{aligned}$$

and for convenience we define $\mathrm{sgn}\,(0) = \mathrm{Un}\,(0) = 0$. This notation extends to $\mathbb{C}$ via the removal of all $\mathbf{j}$, $\mathbf{k}$ terms.

The ranges of most indices x_i, y_m etc. are implicit and follow the convention $i, j, k, l \in \mathbb{Z}_N$, $m \in \mathbb{Z}_{d_H}$, $p \in \mathbb{Z}_{d_L}$ and $q, r, s, t \in \{I, J, K\}$; where N is the size of the training set, d_L the dimension of input space and d_H the dimension of the feature space. Ordering of indices of a dissimilar type (eg. i and q) is arbitrary, so for example $\alpha_{i,q} = \alpha_{q,i}$. Ordering of similar indices, however, is not arbitrary, so for example $Q_{i,j} \neq Q_{j,i}$ in general. Unless otherwise stated summation ranges are implicit, so for example:

$$\sum_i f_i = \sum_{i \in \mathbb{Z}_N} f_i$$

Column vectors will be written in lower case bold (eg. $\mathbf{a}$, $\mathbf{c}$, $\boldsymbol{\alpha}$) and matrices in upper case bold (eg. $\mathbf{A}$, $\mathbf{K}$). Transposition will be indicated by a superscript T (eg. $\mathbf{a}^T$), and conjugate transposition by a superscript $\dagger$ (so $\mathbf{a}^\dagger = \bar{\mathbf{a}}^T$). We denote the identity matrix $\mathbf{I}$, and a vector whose every element is 1 by $\mathbf{1}$. For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{X}^n$, $\mathbf{a}\mathbf{b}^T$ is the outer product, $\mathbf{ab}$ the elementwise product and $\mathbf{a}^T\mathbf{b}$ the inner product. We also define the elementwise norm ($\mathbf{a} = |\mathbf{b}|$), the elementwise sigmoid ($\mathbf{a} = \mathrm{sgn}\,(\mathbf{b})$) and the elementwise unit ($\mathbf{a} = \mathrm{Un}\,(\mathbf{b})$). The standard vector 2-norm is denoted $\|\mathbf{a}\|$ (so $\|\mathbf{a}\| = \sqrt{\mathbf{a}^\dagger\mathbf{a}}$).

Vapnik's ϵ-insensitive risk function [1] (where $\epsilon \in \mathbb{R}\backslash\mathbb{R}^-$) is defined to be:

$$|\vartheta|_\epsilon = \begin{cases} |\vartheta| - \epsilon & \text{if } |\vartheta| \geq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

II. The Standard ϵ-SVR Regression Problem

The standard (real valued) ϵ-insensitive support vector regression (ϵ-SVR) problem is formulated as follows. Given a training set:[1]

$$\begin{aligned}
\Theta &= \{(\mathbf{x}_0, z_0), (\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_{N-1}, z_{N-1})\} \\
\mathbf{x}_i &\in \mathbb{R}^{d_L} \text{ is the } i^{\text{th}} \text{ input vector.} \\
z_i &\in \mathbb{R} \text{ is the system output given input } \mathbf{x}_i.
\end{aligned}$$

where $z_i = \hat{g}(\mathbf{x}_i) + \text{noise}$ for some $\hat{g} : \mathbb{R}^{d_L} \to \mathbb{R}$, and $\mathbf{x}_i$ is drawn from an unknown distribution, construct an approximation $g : \mathbb{R}^{d_L} \to \mathbb{R}$ of $\hat{g}$. An approximation g constructed for a given training set Θ is called a trained machine, and the construction process training. We assume that all noise sources are smooth, independent and identically distributed (i.i.d.) with zero mean.

In the standard approach [14], [1], [15] we implicitly define a nonlinear map $\boldsymbol{\varphi} : \mathbb{R}^{d_L} \to \mathbb{R}^{d_H}$ from input space to feature space, where often $d_H \gg d_L$. Using this map, the trained machine is defined to be:

$$g(\mathbf{x}) = \boldsymbol{\varphi}(\mathbf{x})^T \mathbf{w} + b$$

where $\mathbf{w} \in \mathbb{R}^{d_H}$ is the weight vector and $b \in \mathbb{R}$ is the bias. This is known as the *universal approximator form*, and is a linear function of position in feature space which is nonlinear in input space by virtue of $\boldsymbol{\varphi}$ if $\boldsymbol{\varphi}$ is nonlinear.

The variables $\mathbf{w}$ and b are chosen by solving the ϵ-SVR primal, namely:

$$\min_{\mathbf{w},b} R(\mathbf{w}, b) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \tfrac{C}{N}\sum_i |g(\mathbf{x}_i) - z_i|_\epsilon \tag{2}$$

where R is known as the regularized risk function. The second term in (2) is a measure of the empirical risk associated with this model when it is applied to the training set Θ, and the first term is a regularization term included to minimize sensitivity to noise in $\mathbf{x}$ (as $\nabla_{x_p} g(\mathbf{x}) \propto \|\mathbf{w}\|$). The constant $C \in \mathbb{R}^+$ is used to trade-off empirical risk minimization (and possible overfitting if C is too large) and regularization (and possible underfitting if C is too small), while $\epsilon \in \mathbb{R}\backslash\mathbb{R}^-$ adds a degree of noise insensitivity. The dual form of this optimization problem [3] is usually solved instead of the primal, namely:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} Q = \tfrac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \epsilon|\boldsymbol{\alpha}|^T\mathbf{1} - \boldsymbol{\alpha}^T\mathbf{z} \\
&\text{such that:} \quad -\tfrac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha} \leq \tfrac{C}{N}\mathbf{1} \\
&\qquad\qquad \mathbf{1}^T\boldsymbol{\alpha} = 0
\end{aligned}$$

where $\mathbf{K} \in \mathbb{R}^{N \times N}$, $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $K : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \to \mathbb{R}$ is the kernel function $K(\mathbf{x}, \mathbf{y}) = \boldsymbol{\varphi}(\mathbf{x})^T\boldsymbol{\varphi}(\mathbf{y})$, which may be any function satisfying Mercer's condition [16].

It may be shown that $\mathbf{K}$ is positive semi-definite, and so the dual is a constrained convex quadratic programming problem, and hence all local minima will be global. Note that the exact form of the feature map $\boldsymbol{\varphi}$ is never required for a given Mercer kernel for either training or use, as the construction of the dual requires only the kernel K and the trained machine may be expressed solely in terms of $\boldsymbol{\alpha}$, b and the kernel function [3]:

$$g(\mathbf{y}) = \sum_{i \in SV} K(\mathbf{y}, \mathbf{x}_i)\,\alpha_i + b$$

III. The $\epsilon_{\mathbb{X}}$-SVR Regression Problem

We now consider the extension of the standard ϵ-SVR to complex and quaternionic ϵ-insensitive support vector

[1] Technically, there is no reason to restrict the input space to $\mathbb{R}^{d_L}$ (any Lebesgue measurable set could be used). However doing so would make the construction of the primal less intuitive, and moreover once the dual is attained this extension may be directly introduced using standard extensions to Mercer's theorem.

regression ($\epsilon_{\mathbb{X}}$-SVR), where $\mathbb{X} \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$ is the division algebra of interest. The training set is defined to be:

$$\begin{aligned} \Theta &= \{(\mathbf{x}_0, z_0), (\mathbf{x}_1, z_1), \ldots, (\mathbf{x}_{N-1}, z_{N-1})\} \\ \mathbf{x}_i &\in \mathbb{R}^{d_L} \text{ is the } i^{\text{th}} \text{ input vector.} \\ z_i &\in \mathbb{X} \text{ is the system output given input } \mathbf{x}_i. \end{aligned}$$

where, as for standard ϵ-SVR, $z_i = \hat{g}(\mathbf{x}_i) + \text{noise}$ and we aim to construct an approximation g of $\hat{g}$. In this case, however, $g, \hat{g} : \mathbb{R}^{d_L} \to \mathbb{X}$, and the trained machine is defined by:

$$g(\mathbf{x}) = \boldsymbol{\varphi}(\mathbf{x})^{\dagger} \mathbf{w} + b \tag{3}$$

where $\mathbf{w} \in \mathbb{X}^{d_H}$ is the weight vector, $b \in \mathbb{X}$ is the bias, and $\boldsymbol{\varphi} : \mathbb{R}^{d_L} \to \mathbb{X}^{d_H}$ is the map from input space to (non-real in general if $\mathbb{X} \neq \mathbb{R}$) feature space.

By analogy with the standard ϵ-SVR method, $\mathbf{w}$ and b are chosen to minimize the regularized cost function:

$$\min_{\mathbf{w},b} R_{\mathbb{X}}(\mathbf{w}, b) = \tfrac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} + \tfrac{C}{N}\sum_i |\,|g(\mathbf{x}_i) - z_i|\,|_{\epsilon} \tag{4}$$

where $C \in \mathbb{R}^+$ and $\epsilon \in \mathbb{R} \backslash \mathbb{R}^-$ are constants. This is directly analogous to the standard ϵ-SVR risk, except that:

- The ranges of $\mathbf{w}$ and b differ.
- A complex/quaternionic norm is present inside the ϵ-insensitive risk term so that only the magnitude of the error $z_i - g(\mathbf{x}_i)$ is penalized.

The regularization term $\frac{1}{2}\mathbf{w}^{\dagger}\mathbf{w}$ is included to minimize the sensitivity of the regressor to noise in $\mathbf{x}$. To understand this choice in the current context, let $\hat{\mathbf{w}} = \|\mathbf{w}\|^{-1}\mathbf{w}$ be the unit vector in the direction of $\mathbf{w}$. It may be seen that:

$$\nabla_{x_p} g(\mathbf{x}) = \|\mathbf{w}\| \left(\nabla_{x_p} \boldsymbol{\varphi}(\mathbf{x})\right)^{\dagger} \hat{\mathbf{w}}$$

is the projection of $\nabla_{x_p}\boldsymbol{\varphi}(\mathbf{x})$ in the direction of $\mathbf{w}$, scaled by the magnitude $\|\mathbf{w}\|$ of $\mathbf{w}$. It follows that $\left|\nabla_{x_p} g(\mathbf{x})\right| \propto \|\mathbf{w}\|$, and hence minimizing $\frac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} = \frac{1}{2}\|\mathbf{w}\|^2$ will minimize the sensitivity of the regressor to noise in the input, $\mathbf{x}$.

A. The $\epsilon_{\mathbb{X}}$-SVR Primal

To simplify (4) we use the standard ϵ-SVR approach of introducing a set of slack variables $\boldsymbol{\xi}^* \in \mathbb{R}^N$ into the problem, allowing (4) to be re-expressed as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \; & R_{\mathbb{X}} = \tfrac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} + \tfrac{C}{N}\sum_i \xi_i^* \\ \text{such that:} \; & |g(\mathbf{x}_i) - z_i| \leq (\epsilon + \xi_i^*) \; \forall i \\ & \boldsymbol{\xi}^* \geq \mathbf{0} \end{aligned} \tag{5}$$

Next, to remove the complex/quaternion norm from the constraint set we introduce a set of unit slack variables $\boldsymbol{\alpha}^{\angle} \in \left(S_1^0(\mathbb{X})\right)^N$ to project the error terms $g(\mathbf{x}_i) - z_i$ onto the real axis, and an additional set of slack variables $\boldsymbol{\xi} \in \mathbb{R}^N$ to counter the ambiguity in the sign of this projection. We also re-express the bias using $b = b^{\angle} b^{||}$, where $b^{||} = |b| \in \mathbb{R}$ and $b^{\angle} = \text{Un}(b) \in S_1^0(\mathbb{X}) \cup \{0\}$, to obtain the non-convex quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w},b^{||},\boldsymbol{\xi},\boldsymbol{\xi}^*,\boldsymbol{\alpha}^{\angle},b^{\angle}} \; & R_{\mathbb{X}} = \tfrac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} + \tfrac{C}{N}\sum_i (\xi_i + \xi_i^*) \\ \text{such that:} \; & \text{Re}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) \geq -(\epsilon + \xi_i) \; \forall i \\ & \text{Re}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) \leq (\epsilon + \xi_i^*) \; \forall i \\ & \text{Pu}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) = 0 \; \forall i \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq \mathbf{0} \end{aligned} \tag{6}$$

which will referred to as the $\epsilon_{\mathbb{X}}$-SVR primal. Note that, for any solution, $\xi_i = 0$ and $\xi_i^* = |\,|g(\mathbf{x}_i) - z_i|\,|_{\epsilon}$ if $\text{Re}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) = |g(\mathbf{x}_i) - z_i|$; and $\xi_i^* = 0$ and $\xi_i = |\,|g(\mathbf{x}_i) - z_i|\,|_{\epsilon}$ otherwise. Hence all solutions $\mathbf{w}$, b to (6) must also be solutions to (4) and vice-versa.

The advantage of this form is that it allows us to construct the $\epsilon_{\mathbb{X}}$-SVR dual, which is directly analogous to the standard ϵ-SVR dual. This is desirable for two reasons. Firstly, it makes the feature map implicit rather than explicit, thereby removing any practical constraints on d_H. Secondly, the constraint set of the dual is simpler than the constraint set of the primal, which makes it easier to solve.

B. The $\epsilon_{\mathbb{X}}$-SVR Dual

As the $\epsilon_{\mathbb{X}}$-SVR primal (6) is non-convex it is advantageous to remove this non-convexity before proceeding further. Our method of achieving this is to re-write (6) as a bilevel optimization problem [17]:

$$\begin{aligned} \min_{\boldsymbol{\alpha}^{\angle},b^{\angle}} \; & R_{U\mathbb{X}} = \tfrac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} + \tfrac{C}{N}\sum_i (\xi_i + \xi_i^*) \\ \text{such that:} \; & \text{Pu}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) = 0 \; \forall i \\ & \mathbf{w}, b^{||}, \boldsymbol{\xi}, \boldsymbol{\xi}^* \in \Psi\left(\boldsymbol{\alpha}^{\angle}, b^{\angle}\right) \end{aligned} \tag{7}$$

where $\Psi\left(\boldsymbol{\alpha}^{\angle}, b^{\angle}\right)$ is the set of solutions to the optimisation problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}^{\angle},b^{\angle}} \; & R_{U\mathbb{X}} = \tfrac{1}{2}\mathbf{w}^{\dagger}\mathbf{w} + \tfrac{C}{N}\sum_i (\xi_i + \xi_i^*) \\ \text{such that:} \; & \text{Re}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) \geq -(\epsilon + \xi_i) \; \forall i \\ & \text{Re}\left(\bar{\alpha}_i^{\angle}\left(g(\mathbf{x}_i) - z_i\right)\right) \leq (\epsilon + \xi_i^*) \; \forall i \\ & \boldsymbol{\xi}, \boldsymbol{\xi}^* \geq \mathbf{0} \end{aligned} \tag{8}$$

wherein (7) is referred to as the upper-level $\epsilon_{\mathbb{X}}$-SVR primal problem, $R_{U\mathbb{X}}$ is referred to as the upper-level objective, (8) is referred to as the lower-level $\epsilon_{\mathbb{X}}$-SVR primal problem, and $R_{L\mathbb{X}}$ is referred to as the lower-level objective.

Defining $\boldsymbol{\psi}_i = \boldsymbol{\varphi}(\mathbf{x}_i)\,\alpha_i^{\angle}$ and re-writing (8) in componentwise form the lower-level $\epsilon_{\mathbb{X}}$-SVR primal problem may be written:

$$\begin{aligned} \min_{\mathbf{w}_{\infty},\mathbf{w}_q,b^{||},\boldsymbol{\xi},\boldsymbol{\xi}^*} \; & R_{L\mathbb{X}} = \tfrac{1}{2}\mathbf{w}_{\infty}^T\mathbf{w}_{\infty} + \sum_q \mathbf{w}_q^T\mathbf{w}_q + \tfrac{C}{N}\sum_i (\xi_i + \xi_i^*) \\ \text{such that:} \; & \\ & \boldsymbol{\psi}_{i,\infty}^T\mathbf{w}_{\infty} + \sum_q \boldsymbol{\psi}_{i,q}^T\mathbf{w}_q + \left(\bar{\alpha}_i^{\angle} b^{\angle}\right)_{\infty} b^{||} + \xi_i \geq \left(\bar{\alpha}_i^{\angle} z_i\right)_{\infty} - \epsilon \; \forall i \\ & \boldsymbol{\psi}_{i,\infty}^T\mathbf{w}_{\infty} + \sum_q \boldsymbol{\psi}_{i,q}^T\mathbf{w}_q + \left(\bar{\alpha}_i^{\angle} b^{\angle}\right)_{\infty} b^{||} - \xi_i^* \leq \left(\bar{\alpha}_i^{\angle} z_i\right)_{\infty} + \epsilon \; \forall i \\ & \boldsymbol{\xi} \geq \mathbf{0} \\ & \boldsymbol{\xi}^* \geq \mathbf{0} \end{aligned} \tag{9}$$

which we note is a *convex quadratic* optimisation problem in real variables, and hence has a well-defined dual. To construct this dual, for each of the first set of constraints in the lower-level $\epsilon_{\mathbb{X}}$-SVR primal (9) we associate a Lagrange multiplier $\beta_i \geq 0$, and likewise for each constraint in the second, third and fourth constraint sets in (9) we associate the Lagrange multipliers $\beta_i^* \leq 0$, $\gamma_i \geq 0$ and $\gamma_i^* \geq 0$,

respectively. Using this notation the Lagrangian of (9) is:

$$\begin{aligned}
&\mathcal{L}_{\mathbb{X}} = \tfrac{1}{2}\mathbf{w}_R^T\mathbf{w}_R + \tfrac{1}{2}\sum_q \mathbf{w}_q^T\mathbf{w}_q + \tfrac{C}{N}\boldsymbol{\xi}^T\mathbf{1} + \tfrac{C}{N}\boldsymbol{\xi}^{*T}\mathbf{1} \\
&- \sum_i \beta_i \boldsymbol{\psi}_{i,R}^T\mathbf{w}_R - \sum_i \beta_i \sum_q \boldsymbol{\psi}_{i,q}^T\mathbf{w}_q - b^{||}\boldsymbol{\beta}^T\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}b^{\angle}\right) \\
&+ \boldsymbol{\beta}^T\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}\mathbf{z}\right) - \epsilon\boldsymbol{\beta}^T\mathbf{1} - \boldsymbol{\xi}^T\boldsymbol{\beta} - \boldsymbol{\xi}^T\boldsymbol{\gamma} \\
&- \sum_i \beta_i^* \boldsymbol{\psi}_{i,R}^T\mathbf{w}_R - \sum_i \beta_i^* \sum_q \boldsymbol{\psi}_{i,q}^T\mathbf{w}_q - b^{||}\boldsymbol{\beta}^{*T}\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}b^{\angle}\right) \\
&+ \boldsymbol{\beta}^{*T}\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}\mathbf{z}\right) + \epsilon\boldsymbol{\beta}^{*T}\mathbf{1} + \boldsymbol{\xi}^{*T}\boldsymbol{\beta}^* - \boldsymbol{\xi}^{*T}\boldsymbol{\gamma}^*
\end{aligned} \tag{10}$$

and the Wolfe dual [18] of (9) is:

$$\begin{aligned}
&\min_{\mathbf{w},b^{||},\boldsymbol{\xi},\boldsymbol{\xi}^*} \max_{\boldsymbol{\beta},\boldsymbol{\beta}^*,\boldsymbol{\gamma},\boldsymbol{\gamma}^*} \mathcal{L}_{\mathbb{X}} \\
&\text{such that:} \quad \nabla_{\mathbf{w}}\mathcal{L}_{\mathbb{X}} = \mathbf{0}, \nabla_{b^{||}}\mathcal{L}_{\mathbb{X}} = 0 \\
&\qquad\qquad \nabla_{\boldsymbol{\xi}}\mathcal{L}_{\mathbb{X}} = \nabla_{\boldsymbol{\xi}^*}\mathcal{L}_{\mathbb{X}} = \mathbf{0} \\
&\qquad\qquad \boldsymbol{\beta}, -\boldsymbol{\beta}^*, \boldsymbol{\gamma}, \boldsymbol{\gamma}^* \geq \mathbf{0}
\end{aligned} \tag{11}$$

which has the KKT optimality conditions:

$$\boldsymbol{\gamma}\left(\nabla_{\boldsymbol{\gamma}}\mathcal{L}_{\mathbb{X}}\right) = \boldsymbol{\gamma}^*\left(\nabla_{\boldsymbol{\gamma}^*}\mathcal{L}_{\mathbb{X}}\right) = 0 \tag{12}$$

$$\boldsymbol{\beta}\left(\nabla_{\boldsymbol{\beta}}\mathcal{L}_{\mathbb{X}}\right) = \boldsymbol{\beta}^*\left(\nabla_{\boldsymbol{\beta}^*}\mathcal{L}_{\mathbb{X}}\right) = 0 \tag{13}$$

$$\nabla_{\mathbf{w}}\mathcal{L}_{\mathbb{X}} = \mathbf{0} \tag{14}$$

$$\nabla_{b^{||}}\mathcal{L}_{\mathbb{X}} = 0 \tag{15}$$

$$\nabla_{\boldsymbol{\xi}}\mathcal{L}_{\mathbb{X}} = \nabla_{\boldsymbol{\xi}^*}\mathcal{L}_{\mathbb{X}} = \mathbf{0} \tag{16}$$

$$\boldsymbol{\beta}, -\boldsymbol{\beta}^* \geq \mathbf{0} \tag{17}$$

$$\boldsymbol{\gamma}, \boldsymbol{\gamma}^* \geq \mathbf{0} \tag{18}$$

Condition (16) implies that:

$$\boldsymbol{\beta} = \tfrac{C}{N}\mathbf{1} - \boldsymbol{\gamma} \tag{19}$$

$$-\boldsymbol{\beta}^* = \tfrac{C}{N}\mathbf{1} - \boldsymbol{\gamma}^* \tag{20}$$

and hence, using (17) and (18) and defining $\boldsymbol{\alpha}^{||} = \boldsymbol{\beta} + \boldsymbol{\beta}^*$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{||}\boldsymbol{\alpha}^{\angle}$ it may be seen that:

$$-\tfrac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha}^{||} \leq \tfrac{C}{N}\mathbf{1}$$

It follows from (12) that $\gamma_i = 0$ if $\xi_i > 0$ and $\gamma_i^* = 0$ if $\xi_i^* > 0$ for all i. Combined with (13), (19) and (20), this implies that:

$$\beta_i > 0 \quad \forall i : \mathrm{Re}\left(\bar{\alpha}_i^{\angle}\left(g\left(\mathbf{x}_i\right) - z_i\right)\right) = -\epsilon - \xi_i \tag{21}$$

$$\beta_i = \xi_i = 0 \quad \forall i : \mathrm{Re}\left(\bar{\alpha}_i^{\angle}\left(g\left(\mathbf{x}_i\right) - z_i\right)\right) > -\epsilon \tag{22}$$

$$\beta_i^* = \xi_i^* = 0 \quad \forall i : \mathrm{Re}\left(\bar{\alpha}_i^{\angle}\left(g\left(\mathbf{x}_i\right) - z_i\right)\right) < \epsilon \tag{23}$$

$$\beta_i^* < 0 \quad \forall i : \mathrm{Re}\left(\bar{\alpha}_i^{\angle}\left(g\left(\mathbf{x}_i\right) - z_i\right)\right) = \epsilon + \xi_i^* \tag{24}$$

and hence:

$$\xi_i + \xi_i^* = -\tfrac{N}{C}\left(\left|\alpha_i\right|\epsilon + \mathrm{Re}\left(\bar{\alpha}_i\left(g\left(\mathbf{x}_i\right) - z_i\right)\right)\right) \tag{25}$$

Finally, equations (14) and (15) imply that:

$$\mathbf{w} = \sum_i \alpha_i^{||}\boldsymbol{\psi}_i = \sum_i \alpha_i \boldsymbol{\varphi}\left(\mathbf{x}_i\right) \tag{26}$$

$$\boldsymbol{\alpha}^{||\,T}\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}b^{\angle}\right) = \mathrm{Re}\left(\left(\boldsymbol{\alpha}^{\dagger}\mathbf{1}\right)b^{\angle}\right) = 0 \tag{27}$$

Using these results the lower-level $\epsilon_{\mathbb{X}}$-SVR dual (11) may be written:

$$\begin{aligned}
\max_{b^{||}\in\mathbb{R}} \min_{\boldsymbol{\alpha}^{||}\in\mathbb{R}^N} Q_{L\mathbb{X}} &= \tfrac{1}{2}\begin{bmatrix} \boldsymbol{\alpha}^{||} \\ b^{||} \end{bmatrix}^T \begin{bmatrix} \mathbf{G} & \mathbf{g} \\ \mathbf{g}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha}^{||} \\ b^{||} \end{bmatrix} \\
&\quad + \epsilon\left|\boldsymbol{\alpha}^{||}\right|^T\mathbf{1} - \boldsymbol{\alpha}^{||T}\mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}\mathbf{z}\right) \\
\text{such that:} \quad & -\tfrac{C}{N}\mathbf{1} \leq \boldsymbol{\alpha}^{||} \leq \tfrac{C}{N}\mathbf{1} \\
& \mathbf{g}^T\boldsymbol{\alpha}^{||} = 0
\end{aligned} \tag{28}$$

where $\mathbf{g} \in \mathbb{R}^N$, $\mathbf{g} = \mathrm{Re}\left(\bar{\boldsymbol{\alpha}}^{\angle}b^{\angle}\right)$, $\mathbf{G} \in \mathbb{R}^{N\times N}$, $G_{i,j} = \mathrm{Re}\left(\bar{\alpha}_i^{\angle}K_{i,j}\alpha_i^{\angle}\right)$, $Q_{L\mathbb{X}} = -R_{L\mathbb{X}}$, $K_{i,j} = K\left(\mathbf{x}_i, \mathbf{x}_j\right)$, and $K : \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \to \mathbb{X}$ is the kernel function:

$$K\left(\mathbf{x}, \mathbf{y}\right) = \boldsymbol{\varphi}_m\left(\mathbf{x}\right)^{\dagger}\boldsymbol{\varphi}_m\left(\mathbf{y}\right)$$

which may be any function satisfying a quaternionic extension Mercer's condition [19].[2]

Having expressed the lower-level $\epsilon_{\mathbb{X}}$-SVR training problem in dual form we will now rewrite the upper-level $\epsilon_{\mathbb{X}}$-SVR training problem in terms of $\boldsymbol{\alpha}$ and b, which allows us to merge the resulting bilevel optimization problem back into a standard optimization problem in terms of $\boldsymbol{\alpha}$ and b.

Consider the upper-level $\epsilon_{\mathbb{X}}$-SVR primal (7). Re-expressing in terms of $\boldsymbol{\alpha}$ and b, negating and using (25), (26) and (27) this becomes:[3]

$$\begin{aligned}
\max_{b^{\angle},\boldsymbol{\alpha}^{\angle}} Q_{U\mathbb{X}} &= \tfrac{1}{2}\begin{bmatrix} \boldsymbol{\alpha}^{\angle} \\ b^{\angle} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{H} & \mathbf{h} \\ \mathbf{h}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha}^{\angle} \\ b^{\angle} \end{bmatrix} \\
&\quad + \epsilon\left|\boldsymbol{\alpha}^{||}\right|^T\mathbf{1} - \mathrm{Re}\left(\boldsymbol{\alpha}^{\angle\dagger}\left(\boldsymbol{\alpha}^{||}\mathbf{z}\right)\right) \\
\text{such that:} \quad & \mathrm{Pu}\left(\bar{\boldsymbol{\alpha}}^{\angle}\left(\mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b - \mathbf{z}\right)\right) = \mathbf{0} \\
& \boldsymbol{\alpha}^{||}, b^{||} \text{ solve (28)}
\end{aligned} \tag{29}$$

where $\mathbf{h} \in \mathbb{R}^N$, $\mathbf{h} = \boldsymbol{\alpha}^{||}b^{||}$, $\mathbf{H} \in \mathbb{X}^{N\times N}$, $H_{i,j} = \alpha_i^{||}K_{i,j}\alpha_j^{||}$ and $Q_{U\mathbb{X}} = -R_{U\mathbb{X}}$. Recombining (29) and the lower-level $\epsilon_{\mathbb{X}}$-SVR dual (28) we arrive at the complete, single-level $\epsilon_{\mathbb{X}}$-SVR dual optimization problem:

$$\begin{aligned}
\max_{b,\boldsymbol{\alpha}^{\angle}} \min_{\boldsymbol{\alpha}^{||}} Q_{\mathbb{X}} &= \tfrac{1}{2}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{K} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} \\
&\quad + \epsilon\left|\boldsymbol{\alpha}\right|^T\mathbf{1} - \mathrm{Re}\left(\boldsymbol{\alpha}^{\dagger}\mathbf{z}\right) \\
\text{such that:} \quad & \mathbf{0} \leq \left|\boldsymbol{\alpha}\right| \leq \tfrac{C}{N}\mathbf{1} \\
& \mathrm{Pu}\left(\bar{\boldsymbol{\alpha}}^{\angle}\left(\mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b - \mathbf{z}\right)\right) = \mathbf{0} \\
& \mathrm{Re}\left(\bar{b}^{\angle}\left(\mathbf{1}^T\boldsymbol{\alpha}\right)\right) = 0
\end{aligned} \tag{30}$$

Suppose we neglect the third constraints in (30), leaving b unconstrained. Under this assumption the first order optimality conditions for b is $\nabla_b Q_{\mathbb{X}} = 0$ or, explicitly, $\mathbf{1}^T\boldsymbol{\alpha} = 0$. But this automatically satisfies the third constraint in (30), indicating that this constraint is superfluous. Hence the $\epsilon_{\mathbb{X}}$-SVR dual may be written in a form directly analogous to the

[2]The actual statement of the quaternionic extension of Mercer's condition is essentially identical to the standard statement of Mercer's condition, although more care must be taken with the ordering of elements in the various equations.

[3]When deriving this form, note that constraint (27) allows us to add and subtract arbitrary multiples of $\mathrm{Re}\left(\left(\boldsymbol{\alpha}^{\dagger}\mathbf{1}\right)b^{\angle}\right)$ to $R_{U\mathbb{X}}$ without modifying the resulting expression.

standard ϵ-SVR dual, namely:

$$\begin{array}{ll}\max\limits_{b,\boldsymbol{\alpha}^{\angle}}\min\limits_{\boldsymbol{\alpha}^{||}} Q_{\mathbb{X}} = \frac{1}{2}\boldsymbol{\alpha}^{\dagger}\mathbf{K}\boldsymbol{\alpha} + \epsilon\left|\boldsymbol{\alpha}\right|^{T}\mathbf{1} - \mathrm{Re}\left(\boldsymbol{\alpha}^{\dagger}\mathbf{z}\right) \\ \text{such that:} \quad \mathbf{0} \leq \left|\boldsymbol{\alpha}\right| \leq \frac{C}{N}\mathbf{1} \\ \qquad\qquad \mathbf{1}^{T}\boldsymbol{\alpha} = 0 \\ \qquad\qquad \mathrm{Pu}\left(\bar{\boldsymbol{\alpha}}^{\angle}\left(\mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b - \mathbf{z}\right)\right) = \mathbf{0}\end{array} \tag{31}$$

which reduces to the standard ϵ-SVR dual form if $\mathbb{X} = \mathbb{R}$ (the final constraint is trivially true in this case, as there are no imaginary elements in $\mathbb{R}$).

Note that the form of the $\epsilon_{\mathbb{X}}$-SVR dual is directly analogous to the standard ϵ-SVR problem. It may also be seen, using (3) and (26), that the trained $\epsilon_{\mathbb{X}}$-SVR machine has the form:

$$g\left(\mathbf{y}\right) = \sum_{i \in SV} K\left(\mathbf{y}, \mathbf{x}_i\right)\alpha_i + b \tag{32}$$

and hence, as for standard ϵ-SVR, training and use of the $\epsilon_{\mathbb{X}}$-SVR does not require an explicit feature map, only a kernel function satisfying Mercer's condition.

C. Optimality Conditions

For completeness we now give the optimality conditions of the $\epsilon_{\mathbb{X}}$-SVR dual. To begin we define:

$$\left[\begin{array}{c}\mathbf{e} \\ f\end{array}\right] = \left[\begin{array}{cc}\mathbf{K} & \mathbf{1} \\ \mathbf{1}^{T} & 0\end{array}\right]\left[\begin{array}{c}\boldsymbol{\alpha} \\ b\end{array}\right] - \left[\begin{array}{c}\mathbf{z} \\ 0\end{array}\right]$$

where for all training pairs $(\mathbf{x}_i, z_i) \in \Theta$, $e_i \in \mathbb{X}$ is the difference between the measured output of the system z_i given a input $\mathbf{x}_i$ and the output of the trained machine $g\left(\mathbf{x}_i\right)$ given the same input, Using this notation, the lower-level optimality conditions (21)-(24) may be written:

$$\begin{array}{ll}\left|e_i\right| \geq \epsilon & \forall i : \left|\alpha_i\right| = \frac{C}{N} \\ \left|e_i\right| = \epsilon & \forall i : 0 < \left|\alpha_i\right| < \frac{C}{N} \\ \left|e_i\right| \leq \epsilon & \forall i : \alpha_i = 0\end{array}$$

Combined with the additional constraints of the $\epsilon_{\mathbb{X}}$-SVR dual (30) the optimality conditions may be seen to be:

$$f = 0 \tag{33}$$

$$\mathbf{0} \leq \left|\boldsymbol{\alpha}\right| \leq \frac{C}{N}\mathbf{1} \tag{34}$$

$$e_i = \pm\left(\epsilon + \chi_i\right)\mathrm{Un}\left(\alpha_i\right)\ \forall i : \left|\alpha_i\right| = \frac{C}{N} \tag{35}$$

$$e_i = \pm\epsilon\mathrm{Un}\left(\alpha_i\right)\ \forall i : 0 < \left|\alpha_i\right| < \frac{C}{N} \tag{36}$$

$$\left|e_i\right| \leq \epsilon\, \forall i : \alpha_i = 0 \tag{37}$$

where $\boldsymbol{\chi} \in \left(\mathbb{R}^{+} \cup \{0\}\right)^{N}$.

IV. Experimental Results

In this section we consider a practical application for the $\epsilon_{\mathbb{C}}$-SVR. Specifically, we consider the problem of equalization of a 4-symbol quadrature amplitude modulated (4-QAM) signal over a complex linear communication channel.

All code for this experiment was written in C++, and all simulations were done on a 3.2GHz Pentium D 940 processor based machine with 4GB of RAM running Ubuntu Linux 6.06 (Dapper Drake). The $\epsilon_{\mathbb{X}}$-SVR optimizer code was based on a modified version of SVMHeavy [20].

A. Methodology

We consider equalization of the following channel [10]:[4]

$$\begin{aligned}A\left(z\right) = \frac{o(z)}{s(z)} = \left(0.7409 - \mathbf{i}0.7406\right)\left(1 - \left(0.2 - \mathbf{i}0.1\right)z^{-1}\right) \\ \times\left(1 - \left(0.6 - \mathbf{i}0.3\right)z^{-1}\right)\end{aligned}$$

with additive gaussian noise of variance $\sigma_e^2 = 0.06324$ ($\mathrm{SNR} = 15\mathrm{dB}$); where $s\left(n\right)$ is the channel input sequence and $o\left(n\right)$ the channel output sequence. To facilitate visualization of the problem we have chosen to use a 1-dimensional equalizer and a decision delay of 1. Hence our equalizer should be able to ascertain the input $s\left(n-1\right)$ from the channel output $o\left(n\right)$ (noting that in this experiment we were interested only in $\mathrm{sgn}\left(g\left(\mathbf{x}\right)\right)$).

The channel input is a random complex sequence $s\left(n\right)$ of symbols from the 4-QAM constellation $\{\left(1+\mathbf{i}\right), \left(1-\mathbf{i}\right), \left(-1+\mathbf{i}\right), \left(-1-\mathbf{i}\right)\}$. The channel has 64 output states without noise, which are shown, along with the optimal Bayesian decision boundary for the 15 dB SNR case, in figure 1. For this experiment we set $C = 5$ and $\epsilon = 0.7$ for all experiments, and used a Gaussian RBF kernel $K\left(\mathbf{x}, \mathbf{y}\right) = \exp\left(-\left\|\mathbf{x}-\mathbf{y}\right\|/\gamma\right)$ with $\gamma = 0.7$. We have used a training set containing 1200 random data samples with an SNR of 15 dB. The testing sets used to obtain bit error rates each contained 1 million points with the relevant SNR.

B. Results and Discussion

The decision surface obtained using this training set is shown in figure 2. It can be seen from this that the resulting decision surface represents a good approximation of the Bayesian decision surface in the critical region, as those regions lying within the general vicinity of the 64 noiseless channel outputs are well approximated.

We also compared the symbol error rate of the $\epsilon_{\mathbb{C}}$-SVR equalizer with that of the Bayesian equalizer and two standard ϵ-SVR regressors (one for the real component, one for the imaginary). Results are shown in figure 3. This graph shows that the $\epsilon_{\mathbb{C}}$-SVR outperforms the dual-ϵ-SVR approach and performs only very slightly worse than the optimal Bayesian equalizer.

V. Conclusion

We have proposed the $\epsilon_{\mathbb{X}}$-SVR as a complex/quaternion extension to the standard real-valued ϵ-SV regressor. In contrast to previous approaches, our formulation uses a rotationally symmetric cost function which ensures that the trained machine is influenced only by the magnitude of any training errors and is directionally independent. We have demonstrated that the $\epsilon_{\mathbb{X}}$-SVR outperforms two independent ϵ-SVRs for a 4-QAM channel equalization problem requiring a highly non-linear decision boundary. In addition, our

[4]There is an regrettable notational clash here between the standard mathematical representation of the complex imaginary element, $i = \sqrt{-1}$, the standard engineering representation of the complex imaginary element, $j = \sqrt{-1}$, and the quaternion derived notation, $\mathbf{i} = \sqrt{-1}$. For internal consistency, and as $i, j \in \mathbb{Z}_N$ are used as indexes elsewhere in this paper, we have chosen to use $\mathbf{i} = \sqrt{-1}$ here.

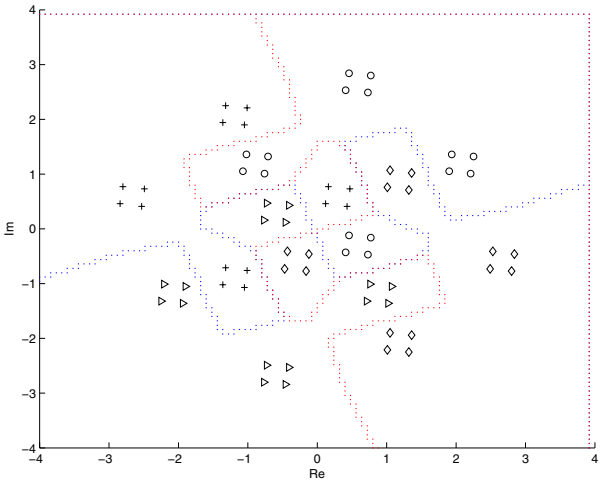

Fig. 1. Bayesian decision boundary. Constellation elements are correctly classified. Key: $+ = 1 + \mathbf{i}$, $\diamondsuit = 1 - \mathbf{i}$, $\triangleright = -1 + \mathbf{i}$, $\circ = -1 - \mathbf{i}$.

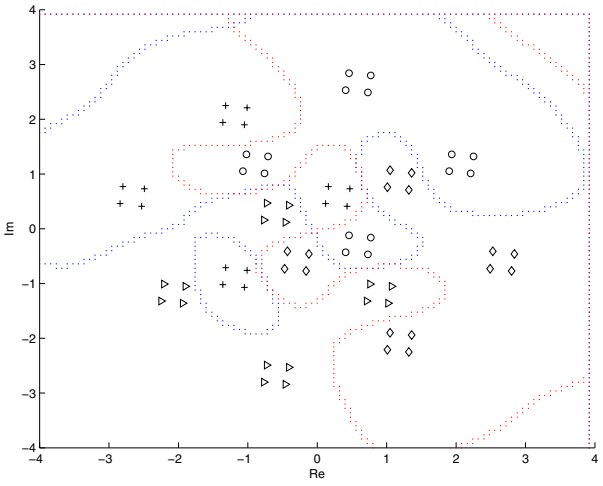

Fig. 2. $\epsilon_{\mathbb{C}}$-SVR decision boundary. Constellation elements are correctly classified. Key: $+ = 1 + \mathbf{i}$, $\diamondsuit = 1 - \mathbf{i}$, $\triangleright = -1 + \mathbf{i}$, $\circ = -1 - \mathbf{i}$.

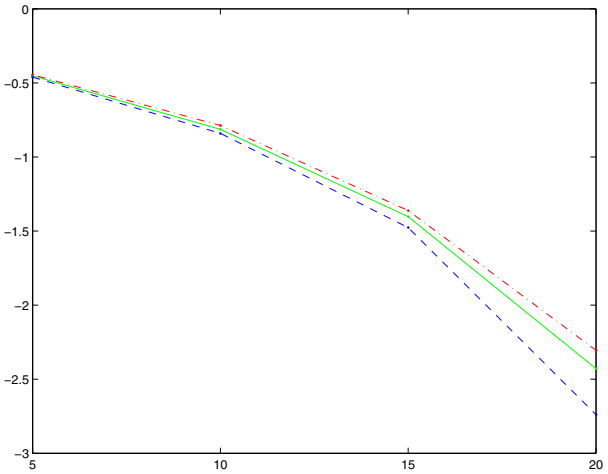

Fig. 3. Symbol error rate curves over varying SNR for $\epsilon_{\mathbb{C}}$-SVR (middle solid curve), dual ϵ-SVR (upper dot-dashed curve) and optimal Bayesian (lower dashed curve) equalizers.

results show that the decision boundary compares favourably with the optimal decision boundary derived using an ideal Bayesian equalizer for this channel over a range of SNRs.

REFERENCES

[1] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, p. 155.

[2] V. Vapnik, S. Golowich, and A. Smola, "Support vector methods for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems*. MIT Press, 1997, vol. 9, pp. 281–187.

[3] A. Smola and B. Schölkopf, "A tutorial on support vector regression," Royal Holloway College, University of London, UK, Tech. Rep. NeuroCOLT2 Technical Report Series, NC2-TR-1998-030, October 1998.

[4] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," in *Advances in Neural Information Processing Systems*, S. Hanson, J. Cowan, and C. Giles, Eds., vol. 5. Morgan Kaufmann, 1993, pp. 147–155.

[6] A. Shilton and M. Palaniswami, "A modified ν-SV method for simplified regression," in *Proceedings of the International Conference on Intelligent Sensing and Information Processing*, 2004, pp. 422–427.

[7] A. Shilton, "Design and training of support vector machines," Ph.D. dissertation, The University Of Melbourne, 2006. [Online]. Available: http://www.ee.unimelb.edu.au/staff/apsh/publications/

[8] E. Bayro-Corrochano and R. Vallejo, "SVMs using geometric algebra for 3D computer vision," in *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-2001)*, 2001, pp. 872–877.

[9] ——, "Geometric neural networks and support multi-vector machines," in *Proc. of IJCNN'2000*, 2000, pp. 389–394.

[10] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, part II: Application to digital communications channel equalisation," *Signal Processing*, vol. 36, pp. 175–188, 1994.

[11] A. Sudbery, "Quaternionic analysis," *Cambridge Philosophical Society*, vol. 85, pp. 199–225, 1979.

[12] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton, 1999.

[13] W. R. Hamilton, "On quaternions: or a new system of imaginaries in algebra," *Phil. Mag. ser. 3*, vol. 25, p. 210, 1845.

[14] A. Smola, "Regression estimation with support vector learning machines," Master's thesis, Technische Universität Münschen, 1996.

[15] B. Schölkopf, P. Bartlett, A. J. Smola, and R. Williamson, "Support vector regression with automatic accuracy control," in *Proceedings of ICANN98, Perspectives in Neural Computing*, L. Niklasson, M. Boden, and T. Ziemke, Eds. Springer Verlag, 1998, pp. 111–116.

[16] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Transactions of the Royal Society of London*, vol. 209, no. A, 1909.

[17] S. Dempe, *Foundations of Bilevel Programming*. Dordrecht: Kluwer Academic Publishers, 2002.

[18] R. Fletcher, *Practical Methods of Optimisation Volume 2: Constrained Optimisation*. Chichester: John Wiley and Sons, 1981.

[19] A. Shilton, "Mercer's theorem for quaternionic kernels," The University Of Melbourne, Department of Electrical and Electronic Engineering, Melbourne, Australia, Tech. Rep. pending, 2007. [Online]. Available: http://www.ee.unimelb.edu.au/pgrad/apsh/publications/

[20] ——, "SVMHeavy: a support vector machine optimiser," 2001. [Online]. Available: http://www.ee.unimelb.edu.au/staff/apsh/svm/