

# Reconhecimento Automático de Modulação Digital de Sinais de Comunicações

Marcelo Corrêa Horewicz, Cairo Lúcio Nascimento Jr. e Waldecir João Perrella

Instituto Tecnológico de Aeronáutica – Praça Marechal Eduardo Gomes, 50 - Vila das Acácias. CEP 12228-900 – São José dos Campos – SP.

**Resumo** — Sistemas de reconhecimento automático de modulação digital devem ser capazes de classificar corretamente os tipos de modulação digital de sinais de comunicações recebidos em um receptor. Este artigo apresenta o resultado da comparação do desempenho dos classificadores de modulação digital baseados na rede neural *feedforward Perceptron* Multicamada (MLP) e os baseados nas *Support Vector Machines* (SVM). Este desempenho foi avaliado por meio de simulações de diferentes tipos de sinais digitais corrompidos com ruído branco aditivo gaussiano (AWGN). É mostrado que a taxa total de sucesso dos classificadores ficou acima de 75% numa relação sinal ruído (SNR) de 5dB.

**Palavras-chaves** — Inteligência Artificial, Redes Neurais, Reconhecimento de padrões, Modulação Digital, MLP, SVM.

## I. INTRODUÇÃO

Reconhecimento automático de modulação digital de sinais de comunicações é um importante problema de processamento de sinais relacionado ao campo das comunicações. Encontra-se num passo intermediário entre a interceptação do sinal e a recuperação da informação, identificando automaticamente o tipo de modulação do sinal recebido para sua posterior demodulação. Possui diversas aplicações, tais como: vigilância eletrônica, avaliação de ameaças, confirmação de sinais, gerenciamento do espectro eletromagnético, utilização em Rádios Definidos por Software (RDS), entre outros.

A discriminação de dados em categorias e classes é denominada de classificação. No caso em que tem-se apenas duas classes, o problema de classificação é conhecido como binário. No entanto, vários problemas reais envolvem a discriminação de mais do que duas classes, sendo referenciados como problemas multiclasse.

Nos problemas de classificação, o principal objetivo é desenvolver um classificador, a partir de um conjunto de dados conhecidos, que seja capaz de determinar corretamente a classe de novos exemplos não conhecidos a priori.

Um classificador de modulação digital é um sistema que automaticamente identifica o tipo de modulação do sinal recebido desde que o sinal exista e que seus parâmetros se encontrem em uma faixa conhecida [1].

Os métodos de reconhecimento automático de modulação digital podem ser divididos em duas principais categorias: Decisão Teórica e Reconhecimento de Padrões [2].

No método de reconhecimento de padrões, os parâmetros considerados relevantes ao problema, conhecidos como *key features*, são representados numericamente e fornecidos como entrada à camada de entrada de um classificador (que pode ou não ser neural).

Este método possui três principais estágios: 1) pré-processamento, na qual as *key features* são selecionadas e extraídas, 2) treinamento e validação, que é a fase de aprendizagem, ou seja, de ajuste da estrutura do classificador, e 3) testes, que é a fase de decisão sobre qual tipo de modulação do sinal fora recebido [3].

Neste artigo é apresentado o resultado da comparação do desempenho de dois tipos de classificadores de modulação digital utilizando reconhecimento de padrões baseados na rede neural tipo *feedforward Perceptron* Multicamada (MLP) e nas *Support Vector Machines* (SVM).

Na seção II é descrita a etapa de pré-processamento dos sinais para a classificação dos classificadores em estudo. Nas seções III e IV algumas considerações teóricas, a configuração e o treinamento dos classificadores MLP e SVM, respectivamente, são apresentados. Na seção V é mostrado o resultado do desempenho destes classificadores no reconhecimento de seis tipos de sinais modulados na fase de testes: ASK2 (*Amplitude Shift Keying*), ASK4, PSK2 (*Phase Shift Keying*), PSK4, FSK2 (*Frequency Shift Keying*), FSK4. Por fim, na seção VI são realizadas as conclusões finais deste trabalho.

## II. PRÉ-PROCESSAMENTO

Os procedimentos para o reconhecimento de sinais digitais foram baseados nos métodos apresentados em [3]. No estágio de pré-processamento o sinal interceptado com  $K$  segundos de tamanho e amostrado na taxa de amostragem  $f_s$ , é dividido em  $M$  sucessivos quadros (*frames*). Cada *frame* possui  $N_s$  amostras resultando em  $M (=kf_s/N)$  *frames*.  $N_s$  foi de 4096 amostras.

Neste trabalho, a frequência de portadora  $f_c$ , a taxa de amostragem  $f_s$  e taxa de símbolos  $r_s$ , foram: 150 Khz, 1200 Khz e 12.5 Khz, respectivamente, resultando em 96 amostras por símbolo ( $f_s / r_s$ ).

Para cada *frame*, um conjunto de *key features* foi extraído para ser fornecido como entrada à camada de entrada de um classificador. Estas *key features* foram derivadas da amplitude instantânea  $A(t)$ , da fase instantânea  $\Phi(t)$  e da frequência instantânea  $f(t)$  do sinal interceptado.

Foram utilizadas sete *key features*, sendo que as cinco primeiras *key features* foram usadas também em [3]:

- 1)  $\gamma_{\max}$  – Máximo valor da densidade espectral de potência da amplitude instantânea central normalizada dos sinais recebidos;
- 2)  $\sigma_{ap}$  – Desvio padrão do valor absoluto da componente não-linear da fase instantânea, avaliada sobre os segmentos relevantes (*non-weak segments*) dos sinais recebidos;
- 3)  $\sigma_{dp}$  – Desvio padrão do valor direto da componente não-linear da fase instantânea, avaliada sobre os segmentos relevantes (*non-weak segments*) dos sinais recebidos;
- 4)  $\sigma_{aa}$  – Desvio padrão do valor absoluto da amplitude instantânea central normalizada dos sinais recebidos;
- 5)  $\sigma_{af}$  – Desvio padrão do valor absoluto da frequência instantânea, avaliada sobre os segmentos relevantes (*non-weak segments*) dos sinais recebidos;
- 6)  $m_a$  - Média da amplitude instantânea central normalizada dos sinais recebidos;
- 7)  $\gamma_{\max f}$  – Máximo valor da densidade espectral de potência da frequência instantânea normalizada dos sinais recebidos.

As entradas da rede foram estas *key features* normalizadas, escalonadas para a faixa de -1 a 1. A normalização foi necessária para tornar o treinamento dos classificadores mais eficiente [4], já que as entradas possuem grandes diferenças de magnitude.

### III. REDE FEEDFORWARD PERCEPTRON MULTICAMADAS (MLP)

#### A. MLP

Uma rede neural artificial (RNA) é um modelo computacional abstrato do cérebro humano. Análogo ao cérebro, uma RNA é composta de neurônios artificiais, chamadas de unidades de processamento, e uma grande interconexão entre estas unidades. Estas redes possuem a habilidade de aprender com exemplos e a generalizar, ou seja, produzir saídas razoáveis para novas entradas que não foram utilizadas no processo de treinamento. As RNAs possuem alguns aspectos que o diferenciam de outras estruturas computacionais que são: aprendizagem por exemplos, habilidade de generalização, a não-linearidade de unidades de processamento, adaptação a novas situações, interconexões paralelas e tolerância a falhas.

Os neurônios em uma rede *perceptron* multicamada (MLP) são dispostos em camadas, contando da camada de entrada (o conjunto de nós de entrada), passando pelas camadas escondidas, até chegar na camada de saída. Ela é uma rede neural global com função de ativação sigmoideal dos neurônios.

Em uma RNA tipo *feedforward* uma unidade de processamento envia sua saída apenas para as unidades das quais ela não recebe nenhuma entrada direta ou indiretamente, implementando um mapeamento estático do seu espaço de entrada para o seu espaço de saída [5].

As *key features* são processadas nas diversas camadas intermediárias (ou ocultas) utilizando-se, para isto, regras de combinação e de ativação. Na camada de saída é gerado o valor  $out_i$ , proveniente da função de saída  $F$  que se refere ao valor da classificação encontrada. Em uma rede com aprendizagem supervisionada este valor encontrado é comparado com o valor desejado e, desta forma, é calculado um erro de estimação, que normalmente é feito ser quadrático. Assim, os diversos pesos da rede (que são

iniciados com valores pré-estabelecidos) são atualizados através de uma regra de aprendizagem.

O algoritmo mais comumente utilizado para atualizar os pesos em uma MLP é o *backpropagation* que utiliza a regra delta generalizada cuja função de custo a ser minimizada é uma função de erro definida pela soma do erro quadrático [6]:

$$E = \frac{1}{2} \sum_{i=1}^k (d_i - out_i)^2 \quad (1)$$

onde  $E$  é a medida do erro total,  $k$  é o número de unidades de saída,  $d_i$  é a  $i$ -ésima saída desejada e  $out_i$  é a  $i$ -ésima saída gerada pela rede. Esta equação define o erro total cometido pela rede.

A regra delta generalizada requer que as funções de ativação utilizadas pelas unidades de processamento sejam contínuas, diferenciáveis e, geralmente, não-decrescentes da entrada total recebida pelo neurônio [6].

A unidade do tipo logística ou tangente hiperbólica é o mais comumente utilizado em aplicações de redes neurais por atenderem aos requisitos da regra delta generalizada. A regra de combinação comumente utilizada é [5]:

$$net_i = \sum_j w_{ij} x_j + bias_i \quad (2)$$

$w_{ij}$  é o peso vindo da unidade  $j$  para a unidade  $i$ ;

$x_j$  é o valor da entrada  $j$ ;

$bias$  ou valor do limiar é um outro peso vindo de uma unidade cuja saída é sempre 1.

Com isto para a unidade tipo logística tem-se [5] e [7]:

$$out_i = sig(net_i) = \frac{1}{1 + e^{(-net_i)}} \quad (3)$$

sendo  $sig(.)$  a função *signal*.

E para a unidade tipo tangente hiperbólica tem-se [5]:

$$out_i = \tanh(net_i) = 2sig(2net_i) - 1 \quad (4)$$

A saída da unidade tipo logística varia entre (0 e 1), enquanto que a saída da unidade tipo tangente hiperbólica varia entre (-1 e 1).

A regra delta generalizada sugere que a variação dos pesos seja definida de acordo com o gradiente descendente do erro com relação ao peso [6]:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i(t) x_j(t) \quad (5)$$

sendo:

$$\delta_i = (d_i - out_i) f'(net_i) \quad (6)$$

o erro se a unidade de processamento  $i$  estiver na camada de saída.

$$\delta_i = f'(net_i) \sum_{l=1}^M \partial_l w_{li} \quad (7)$$

o erro se a unidade de processamento  $i$  estiver em uma das camadas intermediárias.

$\eta$  é a taxa de aprendizado da rede.

### B. Classificador MLP

O classificador de modulação digital baseado na rede *feedforward* MLP foi configurado da seguinte forma: uma camada de entrada com sete unidades, correspondendo as sete *key features* selecionadas, duas camadas escondidas com oito e cinco unidades respectivamente, e uma camada de saída com seis unidades, correspondendo aos seis tipos de modulação a serem classificados, sendo que somente uma unidade de saída deve ser ativada de cada vez, indicando o padrão de entrada classificado. A função de ativação das camadas escondidas escolhida foi a tangente hiperbólica, e a da camada de saída foi a logística.

### C. Treinamento do Classificador MLP

A rede MLP foi treinada com o algoritmo *Levenberg-Marquardt* (LVM) do *Neural Network Toolbox* do MATLAB, por ser um dos mais rápidos algoritmos de treinamento segundo [4]. A rede foi treinada de modo que cada unidade de saída codificasse um tipo de modulação digital. A camada escondida codifica de modo aleatório os mesmos padrões de entrada.

O treinamento da rede foi realizado com 100 amostras de sinais com ruído branco aditivo gaussiano (AWGN) com 20 dB de relação sinal ruído (SNR) e 100 amostras de sinal com 10 dB de SNR para cada tipo de modulação, totalizando 1200 amostras de treinamento, com no máximo 500 épocas.

A quantidade de épocas necessárias no treinamento foi condicionada ao atingimento de um valor limite de erro médio quadrático (MSE), correspondendo a diferença entre o valor desejado e o valor medido no treinamento. Este valor foi estipulado ser de 0.001.

## IV. SUPPORT VECTOR MACHINES (SVM)

O SVM é uma máquina linear que trabalha num espaço de características de alta dimensão formada pelo mapeamento não-linear de um vetor de entrada de dimensão  $N$  em um espaço de características de dimensão  $K$ , sendo  $K > N$ , através de um mapeamento  $\Phi(x)$  [7 - 9].

A arquitetura e a aprendizagem dos parâmetros da rede são tratadas com funções de kernel em vez de processamento direto de sinais das camadas escondidas como no MLP.

As SVMs são baseadas na teoria de aprendizado estatístico introduzido por Vapnik [10]. Ele implementa um mapeamento não-linear executado por um produto interno kernel (polinomial, gaussiano, sigmoidal, entre outros) escolhido a priori dos dados de entrada para um espaço característico de alta-dimensão, em que um hiperplano ótimo é construído para separar os dados linearmente em duas classes, como mostrado na Fig.1. Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço característico é aquele que apresenta a máxima margem de separação. Para dados de treinamento em que as amostras das

diversas classes apresentam superposição (dados não separáveis), uma generalização deste conceito é utilizada, através de um tratamento estatístico.

Embora o SVM separe os dados linearmente em duas classes, o reconhecimento de mais do que duas classes é possível utilizando a estratégia decomposicional que decompõe o problema multiclasse em subproblemas binários. Existem várias técnicas que são utilizadas para resolver problemas de multiclasse, as mais conhecidas são a *one against all* e *one against one*, que foi a utilizada neste trabalho.

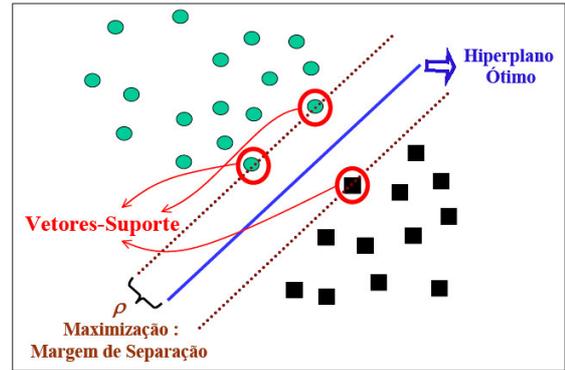


Fig. 1. O hiperplano ótimo separando os dados com a máxima margem  $\rho$ , os vetores-suporte (support vectors) e uma distribuição dos dados no  $R^2$ .

### A. SVMs Lineares

Os SVMs lineares definem fronteiras lineares sobre conjuntos de dados gerais para a separação de dados pertencentes a duas classes, caracterizando um problema de classe binária.

Quando o aprendizado supervisionado é aplicado ao problema de classificação, as amostras de treinamento são formadas pelo conjunto de dados de entrada associados às suas correspondentes respostas pré-classificadas. Após o treinamento, o objetivo é classificar novas amostras, ainda não rotuladas. Considere o conjunto de dados  $T$  de treinamento [8]:

$$(x_i, d_i) \quad 1 \leq i \leq N, \quad x_i \in R^m, \quad d_i \in \{+1, -1\} \quad (8)$$

onde  $x_i$  é o dado de entrada para a amostra  $i$  e  $d_i$  é a correspondente resposta desejada.  $T$  é linearmente separável se é possível separar os dados das classes  $+1$  e  $-1$  por um hiperplano.

Classificações binárias são freqüentemente realizadas pelo uso de funções  $g: x \in R^m \rightarrow R$  com a seguinte estratégia: as amostras são designadas para a classe positiva, se  $g(x) \geq 0$ , e caso contrário, para a classe negativa.

A superfície de decisão será representada por um hiperplano na forma:

$$g(x) = (w^T x) + b = 0, \quad (9)$$

onde  $w \in R^m$  é o vetor de pesos, e  $b \in R$  é o bias.

Assim podemos aplicar a seguinte estratégia de decisão:

$$(w^T x_i) + b \geq 0 \quad \text{para } d = +1 \quad (10.a)$$

$$(w^T x_i) + b < 0 \quad \text{para } d = -1 \quad (10.b)$$

Para descrever o lugar geométrico dos hiperplanos separadores será utilizada a seguinte forma canônica onde o vetor  $w$  e o escalar  $b$  são re-escalados de tal maneira a atender as desigualdades:

$$(w^T x_i) + b \geq +1 \quad \text{para } d = +1 \quad (11.a)$$

$$(w^T x_i) + b \leq -1 \quad \text{para } d = -1 \quad (11.b)$$

Ou de forma mais compacta:

$$d_i [(w^T x_i) + b] \geq 1, \forall (x_i, d_i) \in T \quad (12)$$

Para um dado vetor de pesos  $w$  e bias  $b$ , a separação entre o hiperplano  $g(x) = (w^T x_i) + b = 0$  e o dado de entrada mais perto é chamada de margem de separação denotada por  $\rho$ . Sempre que for possível obter um  $\rho > 0$  existirão infinitos hiperplanos, dentre os quais se busca um hiperplano particular em que a margem de separação  $\rho$  é maximizada.

De acordo com esta condição, a superfície de decisão é dita ser o hiperplano, sendo que os dados de treinamento que se encontram à distância  $\rho$  do hiperplano ótimo são chamados de vetores de suporte (*support vectors*).

Os vetores de suporte exercem um papel importante nas operações deste tipo de aprendizagem de máquina. Em termos conceituais, eles são os pontos que se encontram mais perto da superfície de decisão e, portanto, são os de classificação mais difícil. Como tal, eles têm uma relação direta com a localização da superfície de decisão.

O hiperplano ótimo definido por (12), apresenta um vetor de pesos  $w$  que leva à máxima separação entre as amostras positivas e negativas. Esta condição ótima é alcançada minimizando a norma euclidiana do vetor de pesos  $w$  levando ao seguinte problema de otimização [8]:

$$\text{Minimizar } Q(w, b, \xi) = \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^N \xi_i \right) \quad (13)$$

com as restrições:

$$d_i [(w^T x) + b] \geq 1 - \xi_i \quad \xi_i \geq 0, \forall i=1, \dots, N \in T \quad (14)$$

A constante  $C$  é um termo de regularização que impõe um peso à minimização da complexidade do modelo.  $\xi_i$  são variáveis de folga utilizados para suavizar as margens do classificador linear, permitindo que alguns dados permaneçam entre os hiperplanos e também que ocorra alguns erros de classificação para evitar o superajustamento (*overfitting*).

O problema de otimização obtido é quadrático e pode ser solucionado com a introdução de uma função Lagrangiana como os multiplicadores de Lagrange  $\alpha_i$ .

Desta forma é possível obter o seguinte problema de otimização conhecido como forma dual [7]:

$$\text{Maximizar } Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j d_i d_j (x_i \cdot x_j) \quad (15)$$

$$\text{Com as restrições: } \begin{cases} 0 \leq \alpha_i \leq C, & \forall i = 1, \dots, N \\ \sum_{i=1}^N \alpha_i d_i = 0 \end{cases} \quad (16)$$

## B. SVMs Não Lineares

Na maioria dos casos reais não é possível separar os dados de treinamento por um hiperplano. Nestes casos é preciso realizar um mapeamento de seu espaço original das entradas para um espaço de maior dimensão denominado de espaço de características (*feature space*) [8].

Seja  $\Phi : X \rightarrow \Gamma$  um mapeamento em que  $X$  é o espaço de entradas e  $\Gamma$  é o espaço de características.  $\Phi$  é determinante para que o conjunto de treinamento em  $\Gamma$  possa ser separado linearmente, podendo ser computacionalmente bastante custosa dependendo da dimensão de  $\Gamma$ .

Todas as operações de aprendizagem, treinamento e teste utilizando funções de kernel no SVM satisfazem as condições de Mercer [10]. A função kernel é utilizada no SVM não linear para o cálculo de produtos escalares entre dados no espaço de características:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (17)$$

As funções de kernel mais comumente utilizados são: Polinomiais, RBF (*Radial Basis Function*) ou Gaussianos e os Sigmoidais.

Neste trabalho foi utilizada a função de kernel RBF com parâmetro  $\sigma$ :

$$K(x, x_i) = \exp\left(\frac{-\|x - x_i\|^2}{\sigma^2}\right) \quad (18)$$

Desta forma pode-se alterar (15) para o caso não linear [7]:

$$\text{Maximizar } Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j d_i d_j K(x_i \cdot x_j) \quad (19)$$

com as mesmas restrições de (16).

A solução de (19) leva ao vetor de pesos ótimo da rede SVM [7]:

$$w_{opt} = \sum_{i=1}^{N_s} \alpha_{oi} d_i \Phi(x_i) \quad (20)$$

sendo  $N_s$  o número de vetores de suporte e  $\Phi(x)$  a imagem induzida no espaço de características devido a  $x_i$ .

## C. Algoritmo de Decomposição *One-Against-One*

Suponha uma entrada  $\Gamma_{XY} = \{(x_1, d_1), \dots, (x_i, d_i)\}$  de vetores de treinamento  $x_i \in X \subset \mathbb{R}^n$  e correspondentes saídas  $d_i \in Y = \{1, 2, \dots, c\}$ . O objetivo é treinar as regras de múltipla classificação (*multi-class*)  $q: X \subset \mathbb{R}^n \rightarrow Y$  baseado na estratégia do voto majoritário. A decomposição OAO transforma o problema de múltiplas classificações em séries  $g = (c(c-1)/2)$  de subtarefas que podem ser treinadas por um SVM binário.

## D. Classificador SVM

O algoritmo *One-Against-One*, do *Statistical Pattern Recognition Toolbox* do MATLAB [9], foi utilizado para

implementar o classificador SVM neste trabalho. A função Kernel utilizada foi a gaussiana ( $\sigma=1$ ) e o valor da constante C foi igual a 30. O treinamento foi realizado com 100 amostras de sinais com 20 dB de SNR e 100 amostras de sinal com 10 dB de SNR para cada tipo de modulação, totalizando 1200 amostras de treinamento.

## V. TESTES E RESULTADOS

Na fase de testes, conjuntos de dados de entrada que não foram anteriormente treinados foram apresentados às redes. A taxa de acertos da classificação mostra o desempenho das redes e sua capacidade de generalização.

As taxas de acerto foram calculadas sobre 100 amostras para um conjunto de valores de SNR (25, 20, 15, 10 e 5) em dB e para cada um dos 6 tipos de modulação digital, ou seja em 3000 amostras de testes diferentes dos treinados.

A Tabela I mostra o resultado da taxa de acerto da rede MLP treinada em função do nível de ruído.

**TABELA I TAXA DE ACERTOS DA REDE FEEDFORWARD MLP**

SNR (dB)	Erro Total	Acerto Total	Taxa acerto (%)
25	5	595	99.17
20	9	591	98.5
15	8	592	98.67
10	21	579	96.5
5	150	450	75

No classificador SVM foram realizados 100 testes em cada tipo de modulação e verificado quantas vezes o SVM acertou corretamente a classificação da modulação do sinal de entrada.

A Tabela II mostra o resultado da taxa de acerto da rede SVM treinada em função do nível de ruído.

**TABELA II TAXA DE ACERTOS DA REDE SVM**

SNR (dB)	Erro Total	Acerto Total	Taxa acerto (%)
25	1	599	99.83
20	1	599	99.83
15	0	600	100
10	0	600	100
5	44	556	92.67

A Fig.2 apresenta o teste de generalização, comparando o desempenho dos dois classificadores estudados em função da taxa de acertos de classificação.

Notou-se que o classificador SVM obteve melhor desempenho do que o classificador MLP, principalmente para valores de SNR acima de 5 dB.

## VI. CONCLUSÕES

Reconhecimento automático de modulação digital de sinais de comunicações é um importante problema de processamento de sinais relacionado ao campo das comunicações. Neste artigo, dois classificadores (MLP e SVM) foram empregados para classificar esquemas de modulação digital.

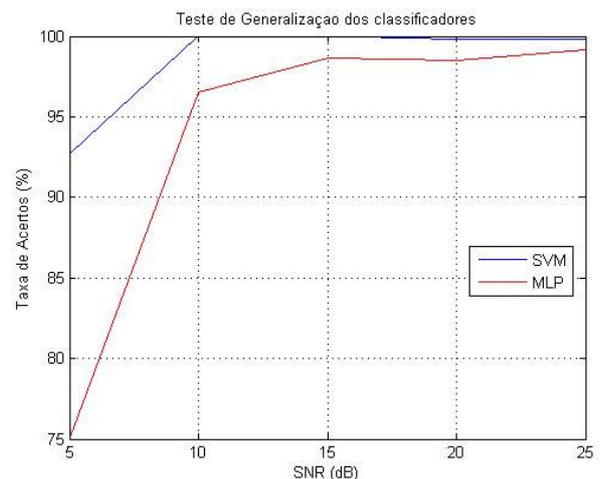


Fig. 2. Teste de Generalização dos classificadores.

Os testes de generalização destes classificadores mostraram que ambos obtiveram uma boa capacidade de generalização para sinais com SNR acima de 10 dB, comprovando a eficiência destas técnicas no problema em questão. Notou-se, que o classificador SVM obteve um melhor desempenho do que o classificador MLP.

Os resultados indicam que estes classificadores podem ser utilizados no reconhecimento de mais esquemas de modulações digitais, necessitando apenas de apropriadas *key features* para isto.

## REFERÊNCIAS

- [1] G. Arlumpalam, V. Ramakonar, A. Bouzerdoum and D. Habibi, "Classification of Digital Modulation Schemes using Neural Networks", Fifth International Symposium on Signal Processing and Its Applications (ISSPA'99), Brisbane, Australia, pp. 649-652, 1999.
- [2] A. Ebrahimzadeh, S.A. Seyedin, "Automatic Digital Modulation Identification in Dispersive Channels", Proceedings of the 5<sup>th</sup> WSEAS International Conference on Telecommunication and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp409-414)
- [3] E.E. Azzouz and A.K. Nandi. *Automatic Modulation Recognition of Communication Systems*. Netherlands: Kluwer Academic Publishers, 1996.
- [4] The MathWorks Inc. *Neural Network Toolbox 5: User's Guide*, 2007.
- [5] Cairo Lúcio Nascimento Júnior e Takashi Yoneyama. *Inteligência Artificial em Controle e Automação*, Editora Wdgard Blücher Ltda, 2000.
- [6] Antônio de Pádua Braga, Teresa Bernarda Luderme e André Carlos Ponce de Leon Ferreira Carvalho. *Redes Neurais Artificiais - Teoria e aplicações*, LTC – Livros Técnicos e Científicos Editora S.A., 2000.
- [7] S. Haykin, *Neural Networks – A Comprehensive Foundation*. Prentice-Hall, 1999.
- [8] Burges, Christopher J.C., *A Tutorial on Support Vector Machines for Pattern Recognition*. *Data Mining and Knowledge Discovery*, 2, 121-167 (1998).
- [9] Franc, V., Hlavac, V., "Statistical Pattern Recognition Toolbox User's Guide", 24 Junho, 2004.
- [10] Vapnik, V. *Statistical Learning Theory*, Wiley, N.Y., 1998.