

REALIDADE AUMENTADA ON-LINE APLICADA À SIMULAÇÃO DE UM MEMBRO SUPERIOR

LUCAS CORREIA DE LIMA ABRAHÃO¹, EDGARD AFONSO LAMOUNIER JUNIOR²

Resumo: Pesquisas em Realidade Virtual e Aumentada aplicada à Engenharia Biomédica têm sido amplamente estudadas nos últimos anos. A simulação de próteses para membros superiores é um assunto amplo e campo de muitas aplicações. Tradicionalmente, próteses virtuais para membros superiores são aplicações criadas para desktops, utilizados para treinamento em centros de reabilitação. Assim, o presente artigo relata sobre o uso de Realidade Aumentada (RA) em uma aplicação web. O objetivo é criar uma página de internet, baseando-se em um ambiente de treinamento para contribuir com a mobilidade de pacientes.

Palavras Chave: Próteses; Realidade Aumentada; Realidade Virtual; Sinal eletromiográfico.

Abstract: Research in Virtual and Augmented Reality applied to Biomedical Engineering has been deeply investigated in recent years. The simulation of upper limb prosthesis is a wide field. Traditionally, virtual upper limb prosthesis is a desktop application, used for training in rehabilitation centers. Thus, this article reports on the use of Augmented Reality (AR) in a web-application. The objective here is to create a home based training environment to help patients with their mobility.

Keywords: Prosthesis; Augmented Reality; Virtual Reality; Electromyographic signal.

¹ lucasabrahao@gmail.com

² elamounier@gmail.com

1 INTRODUÇÃO

1.1 Introdução Teórica

Acidentes acontecem e, infelizmente, pessoas podem perder partes de seus membros. Estudos mostram que nestes casos, a maioria dos indivíduos sofre de maneira física e psicológica. Este é o motivo pelo qual ações para restaurar a liberdade e mobilidade do paciente são imperativas. Soluções tradicionais requerem meios para adaptar o indivíduo a dispositivos protéticos e mecânicos. Estes processos são também aplicados em pacientes que possuem limitações congênitas. Entretanto, um grande esforço mental é necessário por parte dos pacientes durante os primeiros estágios de treinamento. Como consequência, um número significativo destes pacientes desiste de sua recuperação em tempos muito curtos.

O objetivo deste projeto de pesquisa é o de explorar Realidade Virtual e Aumentada na simulação de próteses de membros superiores, por meio de uma solução computacional, para ajudar pacientes durante as primeiras fases de aprendizagem, sem a necessidade de usar as próteses de fato.

Recentemente, a simulação de próteses sintéticas para pacientes amputados ou com deformação congênita tem recebido especial atenção [SOARES ET AL., 2008]. Uma ferramenta muito utilizada nestes simuladores é a Realidade Virtual. Realidade Virtual é um campo da ciência da computação que trabalha com modelagem 3D em um ambiente computacional. De fato, é cada vez mais comum o uso de ambientes virtuais associados ao tratamento reabilitador tanto cognitivo quanto motor, substituindo aos poucos os procedimentos convencionais de tratamento [RIZZO, 2008]. O termo Realidade Virtual (RV) foi criado para definir os ambientes virtuais desenvolvidos com o uso de alta tecnologia para convencer o usuário que ele se encontra em outra realidade. Pode-se dizer que se trata de uma avançada interface homem-máquina que simula um ambiente real. Refere-se também a uma experiência de imersão e interação baseadas em imagens gráficas tridimensionais, geradas por computador.

Nos últimos anos, outra tecnologia que tem apresentado destaque, associado ao crescimento da Realidade Virtual é a Realidade Aumentada (RA). Esta tecnologia permite a inserção de objetos virtuais no ambiente físico, mostrada ao usuário, em tempo real, com o apoio

de algum dispositivo tecnológico, usando a interface do ambiente real, adaptada para visualizar e manipular os objetos reais e virtuais [KIRNER E KIRNER, 2007].

Realidade Aumentada tem provado ser uma ferramenta eficiente para a Saúde e Medicina em aplicações para desktops [SOARES ET AL, 2003; PONS ET AL, 2005]. Para este projeto, o ambiente real é fornecido por uma webcam, e o objetivo é permitir ao usuário interagir com uma prótese de um braço virtual pelo intermédio de diferentes sinais.

Do ponto de vista de reabilitação, uma prótese virtual tem vantagens e desvantagens em relação à prótese real. Mas os objetivos são diferentes: a prótese virtual não substitui a prótese real. Ela permite que o paciente treine com uma prótese, sem causar grandes impactos e dificuldades para o paciente, o que é muito importante em sua reabilitação. Assim, a prótese virtual tem uma primeira grande vantagem: uma pequena quantidade de tempo necessária para fabricá-la. Na verdade, uma prótese real deve ser individualizada para cada paciente, de acordo com sua idade, peso, tamanho e outros fatores importantes. Assim, não é viável utilizar este tipo de prótese, durante o intervalo de tempo em que a prótese de Realidade Aumentada está disponível.

Uma segunda vantagem está relacionada com pacientes que foram vítimas de acidentes. Durante o período de recuperação, é impossível aplicar carga sobre a lesão, ou expô-la ao atrito. É por isso que uma prótese de Realidade Aumentada é útil, pois o contato com o paciente é feito somente, por meio de marcadores e eletrodos, assim problemas como peso, calor e dor não contribuam ainda mais para uma já difícil tarefa, que é a reabilitação.

Por outro lado, a utilização de uma prótese de braço em RA não é fácil. Primeiramente, o usuário precisa ter acesso ao aplicativo. E é por isso, que um dos objetivos deste trabalho, foi criar uma aplicação web, para que o mesmo possa utilizá-la on-line, e assim aumentar a portabilidade e mobilidade deste aplicativo. Em segundo lugar, têm-se dificuldade envolvida na utilização destes aplicativos, e muitas vezes, o usuário é obrigado a ter um período de adaptação antes de utilizar corretamente os mesmos e isso pode gerar uma grande desmotivação por parte do paciente. Esta dificuldade é causada devido ao alto número de ferramentas de processamento de sinal que são utilizados para o controle da prótese real.

Assim, propõe-se investigar técnicas computacionais que permitam a elaboração de um ambiente virtual de RA que dê flexibilidade e independência ao paciente amputado de ter acesso

ao seu treinamento. É importante ressaltar que o intuito deste projeto é criar um sistema apenas no ambiente de treinamento, e não nos aparatos tecnológicos que suportem o mesmo.

1.2 Trabalhos Relacionados

A maioria das pesquisas anteriores utilizava a Realidade Virtual e Aumentada voltada para o controle de sinais eletromiográficos (EMG) de próteses para membros superiores em arquiteturas desktop [SOARES ET AL, 2003; PONS ET AL, 2005; HERLE ET AL, 2008].

Herle [HERLE ET AL, 2008] apresentou um artigo sobre a técnica de classificação de sinais eletromiográficos, baseado em redes neurais. O sistema apresentado em [PONS ET AL, 2005] trabalha com uma prótese de mão controlada via sinais EMG, e apresenta um processo prático e de treinamento para utilização da prótese com maior facilidade. O artigo apresentado em [SOARES ET AL, 2003] focou sobre um mecanismo destinado a ajudar o paciente durante o período de treinamento, já que este pode ser longo e cansativo. Isso resulta em um sistema muito fácil de utilizar e que pode reduzir grandemente o período em estágios de treinamento. O Instituto de Tecnologia de Lund apresentou este artigo [SEBELIUS ET AL, 2005] sobre o controle em tempo real de uma mão virtual. Ele mostra os testes executados em voluntários saudáveis com uma mão virtual, a fim de melhorar a precisão e minimizar o atraso de um conjunto de dez movimentos.

O trabalho em [LAMOUNIER JÚNIOR ET AL, 2010] introduz Realidade Aumentada para a simulação de prótese de membro superior. Para permitir uma melhor imersão do paciente, o autor criou uma aplicação onde os sinais de EMG foram capturados, classificados em um processamento em tempo real, e em seguida, a Realidade Aumentada ajuda o paciente na visualização da prótese.

2 MATERIAL E MÉTODOS

2.1 Metodologia

A aplicação foi baseada na dissertação de mestrado de Kenedy Nogueira [NOGUEIRA, Kenedy, 2007]. Mas, a fim de aumentar a mobilidade dessa aplicação e para que um número maior de pessoas pudesse utilizá-la, precisava-se portar para a internet. A tecnologia Flartoolkit [PONS ET AL, 2005] nos permite programar ambientes de Realidade Aumentada em um aplicativo Flash com ActionScript 3 e linguagem MXML.

Com a ajuda do Flex Builder 3, um objeto *Canvas* foi adicionado e ligado à webcam. Em seguida, a animação do braço foi desenvolvida. Quatro movimentos (pronação, supinação, flexão e extensão) foram projetados em Realidade Aumentada. Normalmente, os objetos 3D são criados em VRML (Virtual Reality Modeling Language). Isso proposto desde que os arquivos que contém os objetos não sejam pesados, e que sejam viáveis para a internet. A figura 1 mostra a versão VRML dos braços em Realidade Virtual. Contudo, o Flartoolkit somente aceita o formato Collada, que é um esquema aberto no padrão digital para aplicações interativas em 3D.

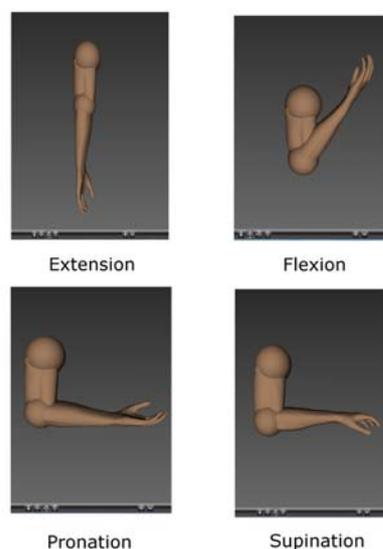


Figura 1: Versão VRML dos quatro movimentos do braço.

A fim de permitir que o usuário possa interagir com o sistema, um algoritmo foi desenvolvido, que utiliza sinais EMG de um arquivo e os envia para o ambiente RA web

definindo um dos quatro movimentos para o qual foi feito (pronação, supinação, extensão e flexão).

Alguns conceitos básicos desta metodologia, bem como as ferramentas utilizadas serão descritas nas próximas seções.

2.2 Sinal Eletromiográfico

O sinal mioelétrico é proveniente do potencial de ação que percorre a fibra muscular levando à contração [Ortolan, 2002].

Existem diferenças de potencial elétrico (DDP) através das membranas (potencial de membrana) de praticamente todas as células do corpo. Contudo, algumas células, como as nervosas e musculares são excitáveis, ou seja, são capazes de gerar impulsos eletroquímicos, e na maioria dos casos utilizam esses impulsos para a transmissão de sinais ao longo das membranas. Essas diferenças de potenciais se dão devido a diferentes concentrações iônicas, dentro e fora das células.

O sinal Eletromiográfico é um reflexo de ativação neuromuscular associada diretamente a uma contração muscular, ou seja, é um sinal de grande complexidade.

Podemos dizer que o sinal eletromiográfico é então, a junção de vários potenciais de ação gerados pelas fibras musculares quando as mesmas são excitadas por neurônios motores, ou seja, é uma manifestação de uma ativação neuromuscular associada a uma contração muscular.

Sua amplitude pode variar de 0 a 10mV. A maior parte da energia útil do sinal é limitada na faixa de frequência entre 0 a 500Hz, com energia dominante entre 50 a 150Hz [LUCA, 1996].

Para se obter um sinal EMG é necessária a soma de milhares de pequenos potenciais elétricos gerados nas fibras musculares, quando excitados. Um exemplo deste tipo de sinal, pode ser visto na Figura 2.

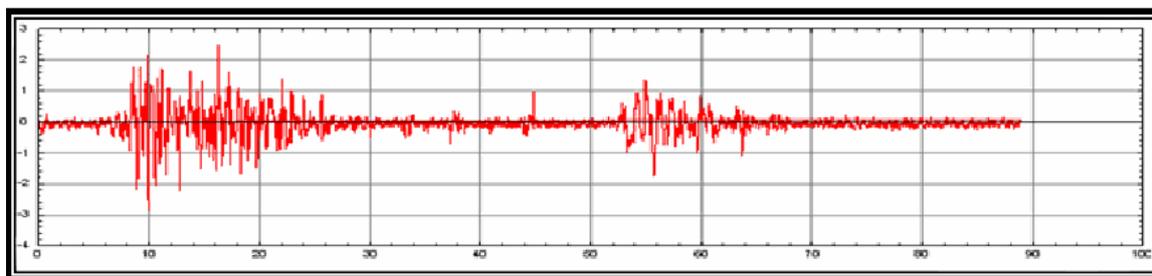


Figura 2: Sinal Eletromiográfico [NOGUEIRA, Kenedy, 2007].

Os potenciais elétricos gerados pelos músculos superficiais que estão se contraindo podem ser captados na superfície da pele por eletrodos de superfície (vide Figura 3), e utilizados no processamento do controle dos atuadores da prótese mioelétrica. O sistema de controle é implementado eletronicamente em um microprocessador dedicado, permitindo ao indivíduo controlar a tensão de alimentação dos sistemas atuadores da mão, do pulso ou do cotovelo, conforme o tipo de prótese. Se há uma contração muscular mais forte, o potencial EMG do músculo é maior, sendo proporcionalmente maior a tensão fornecida ao acionador. Ou seja, a amplitude do sinal de controle é diretamente proporcional à intensidade das contrações, de tal forma que, no caso de uma prótese de mão, por exemplo, a mesma pode ser controlada proporcionalmente nas duas direções (abertura e fechamento). Um impulso muito suave faz com que a mão se mova vagarosamente enquanto que um impulso mais forte faz a mão se mover mais rapidamente [BARROS, 2005].

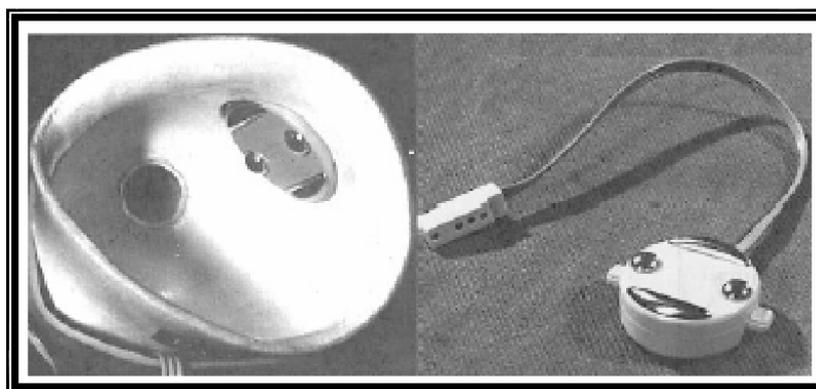


Figura 3: Soquete para braço e eletrodos para uma prótese mioelétrica [ANDRADE, 2000]

Para este estudo, os sinais EMG foram coletados pelos sensores e salvos em documentos texto, para que não haja grande dependência dos pacientes nos testes excessivos da criação do software.

Na detecção do movimento, o sistema proposto deve ser capaz de definir qual movimento será realizado, com base nos sinais EMG coletados. Para isso é necessário extrair um determinado padrão de identificação nos diversos sinais. Adotou-se como sugerido em [ANDRADE, 2000] o uso do algoritmo Auto-Regressivo (AR) de ordem 4.

O método AR utilizado pelo sistema é o algoritmo adaptativo, também conhecido como preditor linear adaptativo. O papel desse estimador ou filtro adaptativo é estimar a amostra atual

com base em amostras de entradas anteriores pela minimização do erro de estimação para cada amostra. Este algoritmo de classificação será melhor explicado na próxima seção.

2.3 O Algoritmo de Classificação

Uma Rede Neural Artificial do (RNA) é um modelo matemático, cuja representação é um grafo orientado. Os nós representam os neurônios artificiais, e as arestas representam as conexões entre os neurônios [ANDRADE, 2000].

O treinamento do RNA é uma adaptação da dissertação de mestrado [NOGUEIRA, Kenedy, 2007]. Dos casos limites (apresentados no trabalho citado anteriormente): os 25 grupos de características são aleatoriamente enviados ao RNA. Após a conclusão processo, os mesmos sinais são enviados para a rede, onde eles podem ser reconhecidos. Com isso, nós pudemos ver que a RNA foi capaz de aprender todos os sinais de todos os grupos de movimentos, treinou-se com a variação de cada limite e a análise de modelos de contração (estático ou dinâmico). Após o cálculo dos coeficientes, o RNA recebe e classifica os movimentos.

Um exemplo de arquitetura RNA com *Multiple Layer Perceptron* (MLP) é mostrada na figura 4. Observe que neste caso, a rede tem uma camada de neurônios na entrada que recebem os coeficientes AR, uma camada interna e uma camada de saída que classifica os movimentos [NOGUEIRA, Kenedy, 2007]. Aqui é o algoritmo usado para resumir os dados coletados a partir do sinal de quatro pontos, e em seguida, escolher uma característica do movimento dos pontos.

- 1- Inicializar todos os coeficientes do filtro $a_m(n)$ para 0;
- 2- Repetir todos os pontos 3 até 5 para cada N para $y(n), n = 0, \dots, N - 1$;
- 3- Calcular o valor estimado do sinal $y(n)$:

$$\hat{y}(n) = \sum_{m=1}^M a_m(n) \cdot y(n - m)$$

- 4- Calcular o erro estimado $e(n)$:

$$e(n) = y(n) - \hat{y}(n)$$

- 5- Atualizar os coeficientes do modelo AR:

$$a_m(n - 1) = a_m(n) - 2\mu e(n) \cdot y(n - m)$$

Onde μ com $0 < \mu < 1$, é uma convergência constante do filtro.

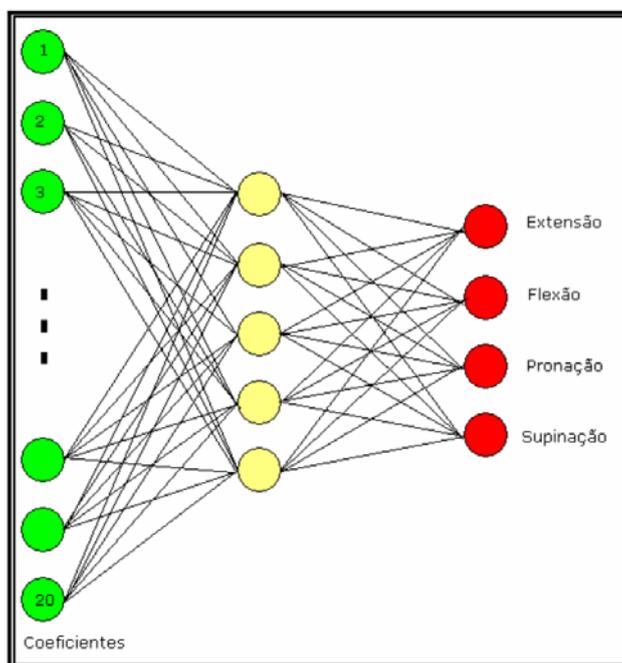


Figura 4: Rede Neural Artificial com *Multiple Layer Perceptron*

Após o cálculo dos coeficientes AR, este alimenta uma Rede Neural Artificial que classifica cada movimento.

A figura 5 mostra como dois diferentes movimentos (flexão e extensão) possuem sinais totalmente diferentes, o que nos permite identificar um movimento com a ajuda do EMG associado ao sinal captado.

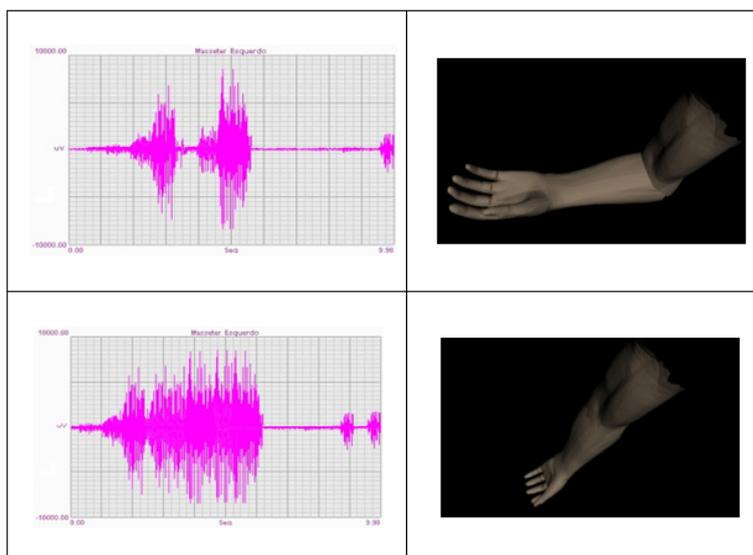


Figura 5: Diferença entre os sinais de identificação de um movimento

2.4 ARToolkit

É um sistema que providencia técnicas de Visão Computacional para calcular a posição e orientação de uma câmera em relação a marcas (em cartões), para que objetos virtuais em 3D possam ser sobrepostos precisamente através das marcas. Primeiro, a imagem real de vídeo é transformada em imagem binária. Esta imagem é depois analisada, de forma a encontrar regiões quadradas na cena real. O *ARToolkit* encontra todos os quadrados na imagem binária. Para cada quadrado, o padrão dentro do quadrado é capturado e comparado novamente com padrões pré-cadastrados [LAMB, 2009].

Numa segunda fase, o *ARToolkit* usa o tamanho do quadrado reconhecido e a orientação de padrões para calcular a posição da câmera de vídeo real em relação à marca física. Uma matriz é preenchida com as coordenadas em relação ao cartão da câmera real. Uma vez que as coordenadas da câmera real e virtual são as mesmas, os gráficos que são desenhados sobrepõem precisamente às marcas reais. Todo processo de funcionamento do *ARToolkit* pode ser visualizado na Figura 6.

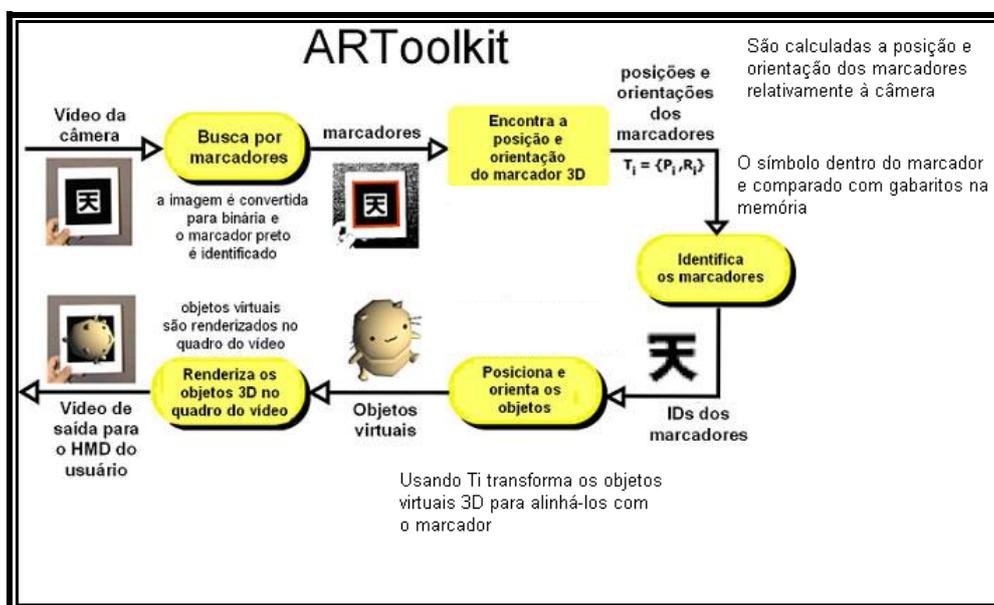


Figura 6 – Esquema de funcionamento do *ARToolkit* [LAMB, 2009]

2.5 Flartoolkit

Um significativa limitação do *ARToolKit* é sua dependência de dispositivos conhecidos como “desktops” (computadores de mesa). Isto significa que para um paciente ou usuário usar o ambiente de treinamento, seu computador estará limitado a um único tipo de sistema operacional (MS Windows, Linux etc.) e todo o software necessário para o funcionamento do *ARToolKit* tem que estar instalado em sua própria máquina. Além disso, onde quer que o usuário queira utilizar o sistema, ele deverá transportar todo seu maquinário ou reinstalar todo o software em outro computador de mesa. Tudo ficaria mais fácil se ele pudesse usar o sistema a qualquer hora, em qualquer lugar e sem dependência de sistemas operacionais e reinstalação de software.

Recentemente, foi desenvolvida, através de uma equipe [PONS ET AL, 2005], uma tecnologia que suporta a portabilidade das bibliotecas do *ARToolKit* para o Adobe Flash[®], trazendo assim Realidade Aumentada para um *Web browser*. Esta tecnologia é conhecida como *FLARToolKit*. Todos os princípios de funcionamento do *ARToolKit* foram preservados, com a vantagem de que agora o usuário precisa apenas ter acesso à Internet (também conhecida como arquitetura em três camadas) e uma webcam (a maioria dos notebooks hoje já são comercializados com as mesmas embutidas). Toda a independência de sistemas operacionais e de software de instalação é garantida. É hoje uma referência para o uso de Realidade Aumentada via web.

FLARToolKit é uma biblioteca de código-fonte aberto para Realidade Aumentada em Flash. Com a ajuda de um marcador como o mostrado na Figura 7 e uma webcam, o objeto virtual pode aparecer no meio ambiente. Um ponto particularmente interessante é que o objeto se adapta ao plano de inclinação e a distância do marcador.

Esta é a chave da Realidade Aumentada, uma vez que nos faz acreditar que o objeto se comporta como um objeto real. Por exemplo, se você colocar o marcador mais próximo da câmera, o objeto irá aumentar, da mesma forma que um objeto amplia quando ele está mais próximo de alguém.

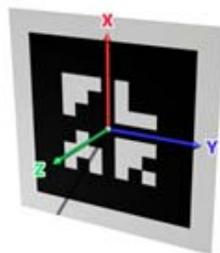


Figura 7: Exemplo de marcador utilizado em Realidade Aumentada

2.6 Arquitetura desktop *versus* arquitetura Internet

Essencialmente, a diferença entre a arquitetura desktop (duas camadas) e a arquitetura Internet (três camadas) está na distribuição de seu software. No modelo duas camadas, a interface de acesso ao usuário e o aplicativo computacional ficam instalados em um mesmo bloco, enquanto os dados ficam armazenados em outro bloco (arquivo ou banco de dados). Isto significa que:

- 1) Atualizações no aplicativo implicam no deslocamento físico de um profissional para reinstalação de novas versões na máquina do cliente.
- 2) A interface de acesso ao aplicativo é dependente da tecnologia de desenvolvimento do mesmo (sistema operacional, linguagens de programação etc.).

Na arquitetura Internet (3 camadas), o bloco de controle de aplicativos é retirado da máquina do cliente e colocado em um outro bloco, conhecido como Servidor de Aplicações. Estes se comunicam através de instalação de redes físicas de computadores. Como consequências disto, as atualizações são feitas apenas no Servidor de Aplicações, liberando o computador cliente de necessidade de instalação física *in loco*. Para tanto, a tecnologia de acesso a aplicações via Web browsers foi desenvolvida independente de sistemas operacionais o que hoje permite uma grande flexibilidade de acesso às aplicações por parte dos clientes.

2.7 A arquitetura das aplicações Flex e PHP

Aplicações Flex podem ser caracterizadas como um modelo de arquitetura SOA, onde o Flex é usado para criar o cliente e para ser conectado a dados utilizando serviços.

Para entender esta arquitetura, considere como navegadores entregam as aplicações web. Quando o navegador faz um pedido, o servidor usa uma combinação de conteúdo estático (códigos HTML / CSS / JS) e scripts (esses scripts podem consultar um banco de dados ou chamar outros scripts, mas no final eles têm uma saída HTML / CSS / JS) para preparar uma página. Esta página é carregada e processada pelo navegador.

Quando um novo estado da aplicação está para ser apresentado, o navegador faz um novo pedido e prepara o servidor da página. O cliente “apenas” processa os dados.

Os aplicativos Flex funcionam de forma diferente. O servidor envia o aplicativo compilado Flex (o arquivo SWF) que funciona dentro do navegador com o plugin do Flash Player. Normalmente, esse arquivo SWF detém apenas a lógica de negócio do cliente. Caso seja necessário (a partir de um banco de dados, por exemplo) a aplicação Flex faz um pedido desses dados. O servidor envia apenas os dados (pode ser em XML, JSON, o formato AMF3), e o cliente sabe como representar dados visualmente. O que acontece aqui é uma arquitetura orientada a serviços: a aplicação Flex é o cliente que pode consumir os serviços de dados do servidor. O aplicativo pode alterar o estado sem atualizar a página ou recarregar o arquivo SWF no navegador. O aplicativo é um cliente que pode fazer mais do que “apenas” processar dados. Assim, usando Flex e Flash Player é possível criar praticamente qualquer coisa que faça sentido para implantar na web, de jogos à aplicativos, widgets que estão integrados dentro “clássico” sistema de aplicações web e muito mais (veja Figura 8).

Traduzindo em poucas palavras, Flex é uma interface gráfica que irá interagir com o usuário e a linguagem PHP fará a tarefa de se comunicar com o servidor.

Esta comunicação é necessária porque os dados são enviados de um computador pessoal (usuário) para um servidor remoto. Quando esses dados são salvos, o programa irá acessar esses dados e compilá-lo, fornecendo os resultados para o usuário.

Quando enviamos um arquivo para o servidor, não nos preocupamos em definir qual é o usuário que está enviando ou se ele tem permissão para acessar determinado diretório. Pode-se definir, dentro do código php qual é a pasta e o nome em os dados que serão salvos, sem qualquer

contato direto com o usuário. Este código, não é mostrado quando a aplicação está rodando e sim, atuando de forma invisível, baseando-se na interação do usuário. Assim, existe um aumento na segurança das aplicações e evita complicações futuras.

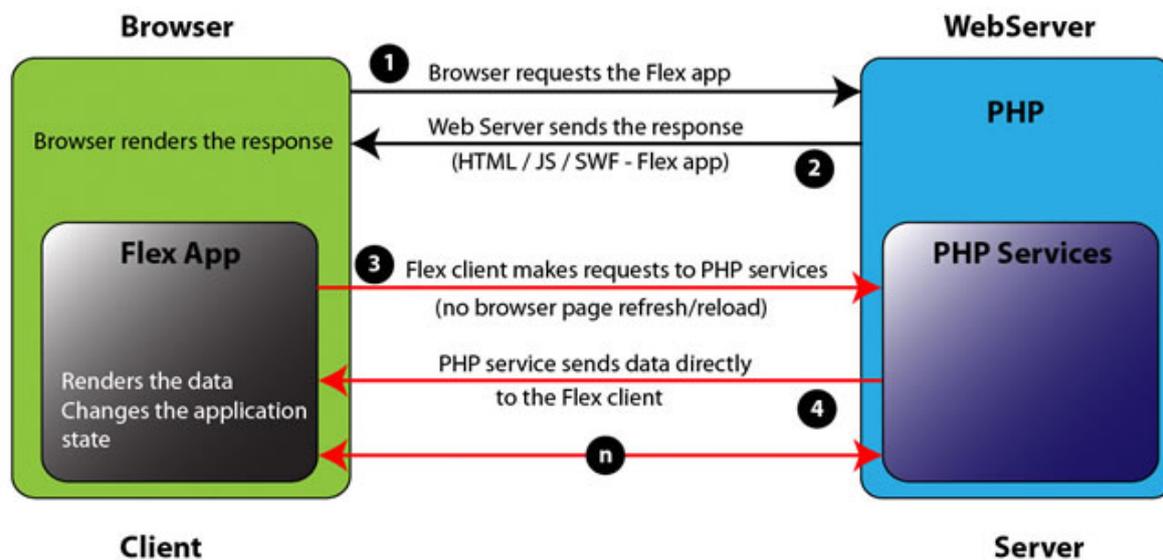


Figura 8: Arquitetura Flex e PHP

3 RESULTADOS E DISCUSSÃO

Para este projeto, vários testes e versões do aplicativo foram criadas. Sempre passo a passo, buscando uma melhoria contínua e de mais fácil manuseio por parte do usuário.

Primeiramente, a figura 9 mostra uma imagem do sistema nos primeiros passos de seu desenvolvimento. O algoritmo de classificação descrito nas seções anteriores deveria ser testado, inserindo os dados já previamente obtidos [NOGUEIRA, Kenedy, 2007]. Pode-se observar na figura, no lado esquerdo, dois botões que permitem fazer o *upload* (envio para o servidor) do sinal associado a um movimento específico. Nesta etapa a função de *upload* ainda estava sendo implementada. A intenção era fazer com que o sinal fosse carregado na área de texto abaixo dos botões. Em seguida, o algoritmo de classificação, fará com que o movimento seja selecionado. Na parte direita da interface, o ambiente virtual é mostrado. É composto pela cena, a webcam grava o objeto virtual sobre o marcador. Também pode-se observar, que nesta primeira etapa, foi utilizado uma garra ao invés de um braço e seu marcador ainda estava com a origem deslocada, tudo para facilitar os testes do algoritmo.

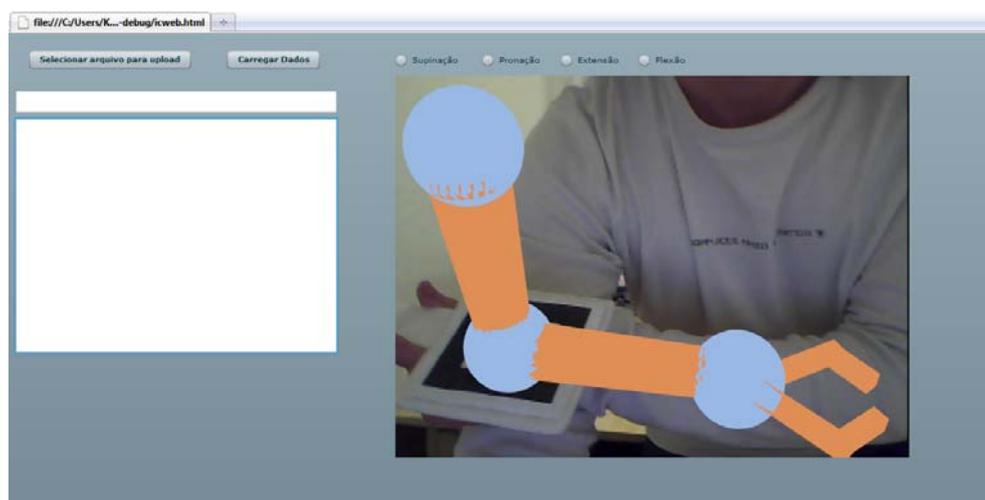


Figura 9: Tela da primeira versão do aplicativo

Com o sucesso nos testes do algoritmo, novas metas foram traçadas e a fim de realizar a arquitetura do sistema proposto, uma nova programação agora iria ser feita. A figura 10 mostra como seria a nova arquitetura do sistema. Primeiramente, o eletrodo recebe o sinal, que o eletromiógrafo irá analisar e envia para o computador. Em seguida, o host executa o aplicativo

em um servidor na internet, que busca no banco de dados para as informações corretas. É então enviado de volta para o aplicativo Flash no computador do host, e a interação com o usuário pode ser visualizada na tela.

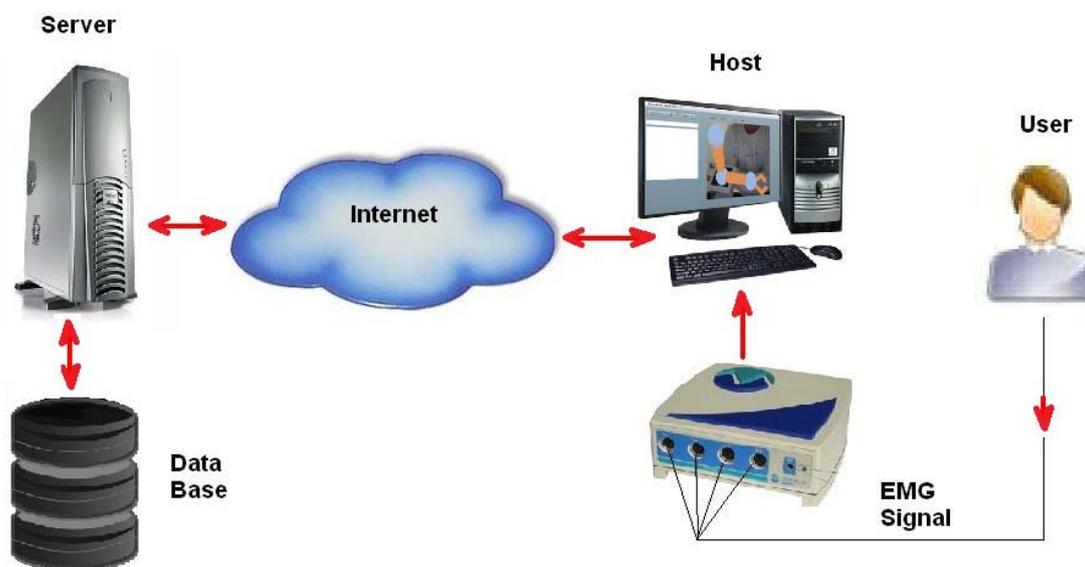


Figura 10: A arquitetura do aplicativo

Com isso, um novo sistema foi criado. Tomando como base esta arquitetura, este sistema recebe e atualiza automaticamente os dados no servidor web, retornando ao usuário o movimento desejado.

Esta nova versão do aplicativo, bem como um exemplo de seu funcionamento, pode ser visto na figura 11. Ela é de fácil manuseio e possui um botão, duas caixas de texto (um para mostrar as variáveis e outra para o nome do movimento) e a cena apresentada pela câmera do usuário.

As grandes diferenças entre as duas versões podem ser vistas nitidamente pela troca da garra mecânica por um braço com texturização neutra, a mudança do ponto de referência para o marcador e a automatização do processo, sendo que agora, o usuário não precisa mais clicar em nenhum botão, pois o sistema capta o sinal salvo pelo sensoriamento e atualiza o movimento automaticamente.

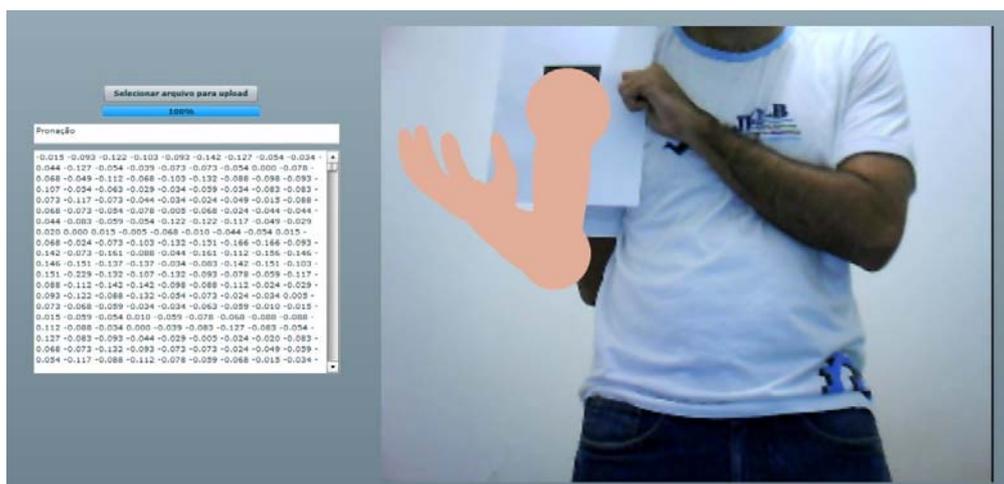


Figura 11: Tela da versão final do aplicativo

Para evitar o desgaste do paciente nos testes para este novo aplicativo, um banco de dados foi coletado e salvo em arquivos de texto (*.txt). Com este banco de arquivos em mãos, criou-se um pequeno software para simular os movimentos aleatórios de um braço, fazendo com que não seja necessária a participação intensa do paciente durante os testes de nosso novo aplicativo web.

Este software, que pode ser visto na figura 12, simula os movimentos (pronação, supinação, extensão e flexão) aleatoriamente, salvando os dados em um arquivo de texto e o atualizando a cada segundo. Esta atualização altera o arquivo criado e salva com os novos dados a cada variação de movimentos.

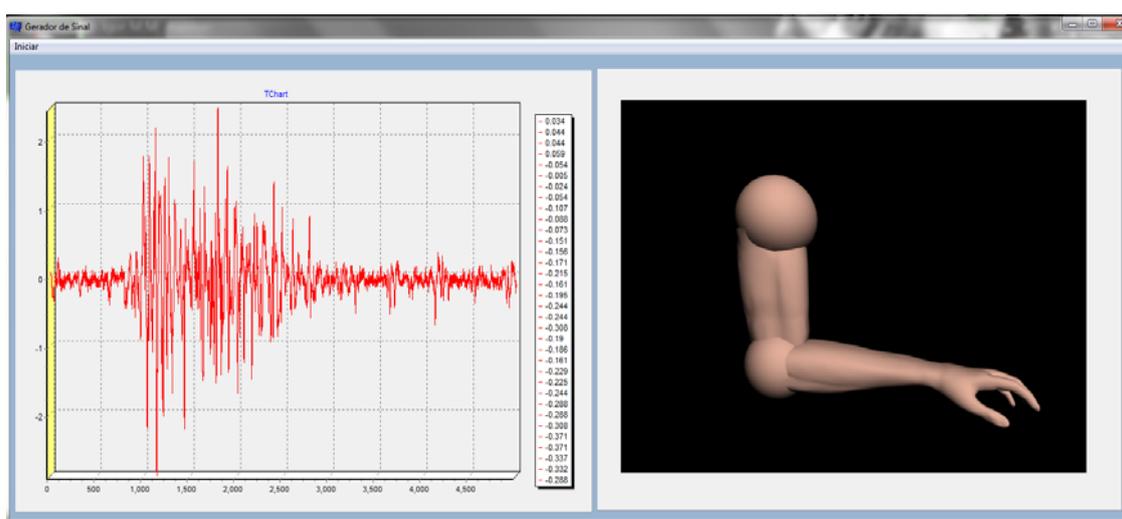


Figura 12: Software de geração de sinal aleatório

Após a criação deste arquivo texto, que no caso do usuário será através da interface de sensoriamento, o mesmo acessa ao aplicativo na internet e clica no único botão disponível para o carregamento do arquivo. Ao clicar, abrirá uma tela, pedindo para que o usuário entre com o endereço do arquivo texto gerado no seu computador. Este simples clique, aciona o algoritmo em PHP que é responsável pelo carregamento do arquivo no servidor (como explicado na seção 2.7), para que este possa receber e trabalhar em cima de seus dados e retornar um valor específico de acordo com o algoritmo.

Uma vez feito este procedimento, não mais o paciente necessitará de clicar neste botão, isso porque o software atualiza automaticamente os dados deste arquivo de segundo em segundo.

Recebendo estes dados via texto, o aplicativo web quebra o conjunto de números salvos (estes números representam ao sinal discretizado) e os ordena em variáveis. Estas variáveis são processadas no algoritmo de classificação (apresentado na seção 2.3) e retorna a imagem em Realidade Aumentada para o usuário, bem como o nome do movimento em uma caixa de texto.

Os testes deste aplicativo foram feitos através de um servidor local, idêntico a um servidor que se é utilizado na internet.

4 CONCLUSÕES

Embora, para se deslocar de uma arquitetura de *desktop* para a web não seja uma tarefa fácil, o sistema proposto tem demonstrado que tal migração aplicada à simulação virtual do membro superior é possível.

Quando pensávamos que seria um sistema inviável, devido ao tempo de reposta do aplicativo, o sistema respondeu bem, sem problemas de atrasos. Isso se deve porque os arquivos que são enviados para o servidor web, são relativamente leves o que não impede a viabilização desse software.

Apesar das dificuldades, soluções foram desenvolvidas e postas em práticas de modo a simplificar o trabalho, tanto para o usuário quanto para o profissional que irá conduzir na recuperação do paciente.

Grandes melhoras ainda podem ser realizadas com futuras pesquisas. Alguns dos futuros trabalhos devem concentrar-se na melhor utilização de técnicas computacionais para melhorar o controle e simulação da prótese virtual. Pode-se melhorar também na interface (layout) de modo que faça com que o usuário se sinta bem ao manusear o aplicativo, assim como um banco de texturizações para os braços, fazendo com que o usuário possa escolher o tipo de braço que melhor se encaixa em seu corpo para uma melhor adaptação com o aplicativo.

Além de tudo, outros movimentos podem ser construídos para o sistema e assim, um programa mais completo e real pode ser criado, facilitando, a vida de muitos pacientes.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, Adriano de Oliveira; Metodologia para Classificação de Sinais EMG no Controle de Membros Artificiais. Uberlândia: Universidade Federal de Uberlândia, Dissertação de Mestrado, 2000.

BECK, K.; Programação extrema explicada: acolha as mudanças. Bookman, Porto Alegre, 2004.

BECK, K., EL AL.; Manifesto for agile software development. Disponível em: <<http://www.agilemanifesto.org>>, 2001. Acesso em 22 de Março de 2010.

CARDOSO, A.; KIRNER, C.; LAMOUNIER, E. e KELNER, J.; Tecnologias para o desenvolvimento de sistemas de realidade virtual e aumentada, Editora Universitária da UFPE, 2007.

COLT, H; CROWFORD, S. and GALBRAITH, O.; Virtual reality bronchoscopy simulation- A revolution in procedural training, Chest, 2001, 120: 1333-1339.

FILHOAIS, C. and TRINDADE, J.; Física no computador: O computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. Revista Brasileira de Ensino de Física, No. 25, 2003.

HERLE, S.; RAICA, P.; LAZEA, GH.; ROBOTIN, R.; MARCU, C.; and TAMAS, L. Classification of surface electromyographic signals for control of upper limb virtual prosthesis using time-domain features. 2008.

KIRNER, C. e TORI, R.; Introdução à Realidade Virtual, Realidade Misturada e Hiper-Realidade. In: (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. São Paulo, 2004. Introdução à Realidade Virtual, Realidade Misturada e Hiper-Realidade, p.3-20

KIRNER, C.; KIRNER, T.G.; Virtual Reality and Augmented Reality Applied to Simulation Visualization. In: El Sheikh, A.A.R.; Al Ajeeli, A.; Abu-Taieh, E.M.O. (Org.). Simulation and Modeling: Current Technologies and Applications. 1 ed. Hershey-NY: IGI Publishing, 2007, v. 1, p. 391-419.

LAMB, P. **ARToolKit Home Page, Documentation, Tutorials.**
<http://www.hitl.washington.edu/artoolkit/documentation/tutorialhistory.htm> - Acesso em 23 de Março de 2010.

LAMOUNIER JÚNIOR, E. A. ; SOARES, Alcimar ; ANDRADE, A. ; CARRIJO, R. . A Virtual Prosthesis Control Based on Neural Networks for EMG Pattern Classification. In: ASC´, 2002, Banff. Proceedings of the Sixth IASTED International Conference on Artificial Intelligence and Soft Computing. p. 425-430. (*Best Paper Award*).

LAMOUNIER JÚNIOR, E. A; LOPES K.; ANDRADE, A. O. CARDOSO, A.; and SOARES, Alcimar. On the use of virtual and augmented reality for upper limb prostheses training and simulation. 2010.

LUCA., C. J. D. Surface electromyography: Detection and recording. Neuromuscular Research Center and Biomedical Engineering Dept. and Neurology Dept., Boston University, Delsys Inc. 1996.

NOGUEIRA, Kenedy. O uso de técnicas de realidade virtual e aumentada na simulação de prótese. Dissertação de Mestrado, Universidade Federal de Uberlândia, 2007.

NOGUEIRA, Kenedy; LAMOUNIER JÚNIOR, E. A.; CARDOSO, A. Upper Limb Prosthesis Training using Augmented Reality. Aipr – International Conference on Artificial Intelligence and Pattern Recognition Orlando, FL, USA, 2007.

PONS, L.; CERES, R.; ROCON, E.; LEVIN, S.; MARKOVITZ, I.; SARO, B.; REYNAERTS, D.; VAN MOORLEGHEM, W.; and BUENO, L. Virtual reality training and emg control of the manus hand prosthesis. Cambridge University Press, 2005.

RIZZO, A.A., communications: “Virtual Reality in Psychology and Rehabilitation: The Last Ten years and Next”. VRIC - 2008 Symposium, 2008.

SAQOOSHA. Flartoolkit. <http://sagoosha.net/en/flartoolkit/>. Acesso em 23 de Março de 2010.

SEBELIUS, F.; AXELSSON, M.; DANIELSEN, N.; SCHOUENBORG, J., and LAURELL. T. Real-time control of a virtual hand. Lund Institute of Technology, 2005.

SOARES, Alcimar ; LAMOUNIER JÚNIOR, E. A. ; LOPES, Kenedy ; ANDRADE, A. AUGMENTED REALITY: A TOOL FOR MYOELECTRIC PROSTHESES. In: ISEK 2008, 2008, Niagara Fall, Canada. XVIIth Congress of the International Society of Electrophysiology and Kinesiology, 2008. v. 1.

SOARES, A.; ANDRADE, A.; LAMOUNIER JÚNIOR, E. A; and CARRIJO, R. The development of a virtual myoelectric prosthesis controlled by an emg pattern recognition system based on neural networks. Journal of intelligent information systems, 2003.

WEIDLICH, D., Scherer, S. and WABNER, M.; “Analyses Using VR/AR Visualization”, IEEE Computer Graphics and Visualization, pp.84-86, September/October 2008.