

# Internet das Coisas Musicais Aplicada a Instrumentos Musicais Robóticos

## (Internet of Musical Things Applied to Robotic Musical Instruments)

Higor Camporez<sup>1</sup>, Anilton Garcia<sup>1</sup>, Jair Silva<sup>1</sup>, Leandro Costalonga<sup>2</sup> e Helder Rocha<sup>1</sup>

<sup>1</sup>*Departamento de Engenharia Elétrica (DEL) – Universidade Federal do Espírito Santo*

<sup>2</sup>*Departamento de Computação e Eletrônica (DCEL) – Universidade Federal do Espírito Santo*

higorcamporez@gmail.com, anilton.garcia@ufes.br, jair.silva@ufes.br, leandro.costalonga@ufes.br, helder.rocha@ufes.br

### Abstract

In the 1950s, musicians started to explore computer use for musical activities, often unrelated to real-time processing. However, technological advances have made the computer more present in musical performances, creating research areas such as ubiquitous music (ubimus), which aims to promote tools that support creative musical activities and the Internet of Musical Things (IoMusT), which is described as the use of electronics (sensors, actuators, connectivity, etc) for a musical purpose. An application encompassing these research areas is the development of robotic instruments for group musical performances. In this paper, some of the key concepts of IoMusT and ubimus are presented alongside examples using musical robots prototypes (RoboMus). In addition, a proposal for robot synchronization including a mechanical delay compensation through neural networks and precision time protocol (PTP) to align the robot's clock.

**Keywords:** *Digital Musical Instrument, Internet of Musical Things, Robotics Musicians, Synchronization, UbiMus.*

### Resumo

Em meados da década de 1950, os músicos começaram a fazer uso do computador em atividades musicais, frequentemente não relacionados a processamento em tempo real. Porém, os avanços tecnológicos propiciaram um maior uso do computador em performances musicais, criando novas áreas de pesquisa como a de música ubíqua (ubimus), que promove ferramentas tecnológicas para auxiliar atividades musicais criativas e a Internet das Coisas Musicais (*Internet of Musical Things - IoMusT*), descrita como o uso da eletrônica (sensores, atuadores, conectividade e etc.) para propósitos musicais. Uma aplicação englobando essas duas áreas de pesquisa é o desenvolvimento de instrumentos musicais robóticos para performances em conjunto, que demandam sincronização. Este artigo apresenta conceitos de IoMusT e ubimus juntamente com exemplos de protótipos de robôs musicais (RoboMus), além de uma proposta para a sincronização dos robôs utilizando redes neurais para aprendizado e compensação de atrasos mecânicos e o protocolo PTP (*Precision Time Protocol*) para alinhamento de relógios.

**Palavras-chave:** *Internet das Coisas Musicais, Instrumentos Musicais Digitais, Música Ubíqua, Músicos Robóticos, Sincronização.*

## 1. Introdução

Em meados de 1950 uma nova tecnologia despertou o interesse dos músicos – o computador. Tão logo foi criado, o computador começou a ser explorado pelos músicos para atividades de composição musical. Com a popularização desse dispositivo, ocorreu um grande progresso no campo da Interação Humano-Computador (IHC) de modo a permitir que os computadores pudessem ser utilizados por pessoas não especializadas. Certamente o fazer musical não figurava entre as

principais utilizações do computador e, sendo assim, a interação como instrumentos musicais não evoluiu tão rapidamente como os demais produtos da computação, como por exemplo, editores de texto. Atualmente, computadores estão envolvidos em praticamente todas as atividades conhecidas por seres humanos, incluindo a música. Dentre essas atividades encontra-se a performance musical, que apresenta certa complexidade devido as suas demandas temporais.

A ampla interseção entre atividades musicais e tecnologia levaram ao nascimento de uma área de pesquisa conhecida como música ubíqua (ubimus – *ubiquitous music*) (Keller, Lazzarini, & Pimenta, 2014); esta área lida com sistemas de agentes humanos e recursos materiais que dão suporte para atividades musicais através de ferramentas de apoio à criatividade. Na prática, ubimus é música (ou atividades musicais) apoiada por conceitos de computação e suas tecnologias. Relaciona-se com conceitos como: portabilidade, mobilidade, conectividade e disponibilidade (incluindo não músicos) (Keller, Flores, Pimenta, Capasso, & Tinajero, 2011).

Outra área de pesquisa fortemente vinculada a ubimus, englobando conceitos tecnológicos para atividades musicais é a Internet das Coisas Musicais (IoMusT) (Keller & Lazzarini, 2017; Turchet, Fischione, Essl, Keller, & Barthet, 2018). A IoMusT, uma subárea de *Internet of Things* (IoT) (Atzori, Iera, & Morabito, 2010), é descrita como o uso da eletrônica por meio de sensores, atuadores, processamento e conectividade de rede para coleta e troca dados visando um propósito musical, permitindo assim, a interconectividade entre artistas e/ou a audiência.

O uso da eletrônica para atividades musicais também tem resultado no desenvolvimento de instrumentos musicais robóticos (Bretan & Weinberg, 2016; Kapur, 2005) que utilizam sensores, atuadores e outros diversos recursos para construção de robôs, entrelaçando assim as áreas de IoMusT e ubimus para esse contexto. Dentre os trabalhos na área, destaca-se um robô interativo construído para tocar marimba, chamado de *Shimon* (Hoffman & Weinberg, 2010). *Shimon* dispõe de componentes que permitem a improvisação com outros músicos em tempo real, de acordo com o contexto, e a interação gestual através de uma cabeça não-humanoide. Outro robô em destaque é o *LEMUR GuitarBot* (Singer, Feddersen, Redmon, & Bowen, 2004), uma espécie de *slide-guitar* com quatro cordas, controlada por mensagens MIDI. Em cada corda encontra-se um dispositivo de *slide* que funciona de forma independente. Os robôs musicais podem funcionar de forma autônoma, como o Shimon, ou podem ser controlados por seres humanos através de interfaces de controle e rede de computadores, como é o caso da plataforma RoboMus (Camporez, Mota, Astorga, Rocha, & Costalonga, 2018), descrita na Seção 4.

Nesse artigo, aborda-se ubimus, IoMusT e seus conceitos, seguido por uma descrição de implementação utilizando robôs musicais (RoboMus) controlados por seres humanos, citando vários exemplos práticos construídos, bem como uma interface de controle. Ainda é apresentada uma proposta de sincronização para esses robôs, incluindo a sincronia de relógios e a compensação de atrasos mecânicos através de redes neurais.

## 2. Contribuições

O presente trabalho apresenta contribuições para as áreas de pesquisa ubimus e loMusT. As principais contribuições são: (a) a implementação de protótipos de robôs musicais e (b) uma proposta de técnicas para sincronização de robôs musicais que inclui sincronia de relógios e compensação dos atrasos mecânicos através do uso de redes neurais.

### 3. Internet das Coisas Musicais

Em uma perspectiva da ciência da computação, a coisa musical (*musical thing*) é definida como um dispositivo computacional capaz de detectar, adquirir, processar, atuar e trocar informações para um propósito musical (Turchet et al, 2018). loMusT pode ser descrito como a conjunção de interfaces, protocolos e a representação de informações musicais que possibilitam serviços e aplicações para finalidades artísticas baseadas em interações entre coisas musicais ou entre humanos e coisas musicais, seja no mundo físico ou digital. Assim, loMusT pode ser retratada como a conjunção de IoT e os sistemas musicais.

Os ecossistemas digitais (Boley & Chang, 2007) também abrangem a área de IoT (Delicato et al, 2013), e, por conseguinte, o campo de loMusT. De acordo com Turchet et al (2018), um ecossistema de loMusT é composto por: usuários envolvidos em atividades musicais; provedores de informações e serviços; e padrões. Os autores também definem os três principais componentes para um ecossistema de loMusT dentro de uma perspectiva tecnológica:

- Coisas musicais: Uma nova classe de dispositivos musicais conectados à Internet que promete uma transformação notória no modo de interação do usuário com atividades musicais, conduzidas com objetos musicais. Essas coisas musicais herdam diversas características de IoT como sensores, atuadores, conectividade e software para manipulação de dados.
- Conectividade: Na infraestrutura de conexão usada em loMusT, deve-se considerar uma comunicação multidirecional sem fio, seja ela remota ou localizada. Essa interconectividade é alcançada por meio das tecnologias de hardware e software, e, também, por padrões e protocolos de gerenciamento. Algumas performances artísticas demandam comunicação em tempo real. Portanto, a rede deve fornecer baixa latência, alta confiabilidade, alta qualidade e sincronização entre os dispositivos conectados.
- Aplicações e serviços: Vários tipos de aplicações e serviços podem ser construídos com essa nova perspectiva sobre IoT, onde o público-alvo são músicos, engenheiros de áudio, membros da audiência e outros. Os serviços podem ter caráter interativo ou não, exemplos são: conectividade do *smartphone* com atividades musicais; serviços de análises para conteúdo criativo; serviço de mapeamento dos dados e controle de coisas musicais. Já as aplicações representam a mais alta camada de interação com o usuário. Para os membros da audiência, pode-se considerar como exemplo o aumento da experiência de um concerto baseado na interação com coisas musicais. Para os músicos, um exemplo é a possibilidade de ensaios remotos. Para engenheiros de áudio encontram-

se os estúdios de produção inteligentes. Para estudantes e professores, tem-se a possibilidade de criar seções remotas de ensino.

De acordo com Turchet et al (2018), loMusT envolve diversas áreas como: performances musicais em rede; performances interativas; dispositivos vestíveis; realidade aumentada e etc. Como exemplos práticos de loMusT, pode-se citar:

- Dispositivos vestíveis: Aparato alocado junto ao corpo do usuário (ex. *smartwatch*) com objetivo de captar informações fisiológicas como temperatura corporal, frequência cardíaca e captação de movimentos. Podendo também fornecer uma resposta háptica de acordo com as informações captadas e o contexto da performance;
- Robôs musicais: instrumentos musicais robóticos, podendo ser robôs virtuais ou físicos. Os físicos fazem uso sensores e atuadores, ou seja, coisas musicais. Na classe de robôs controlados por seres humanos, em geral, demanda-se de conexão, representando o componente conectividade e interfaces de controle (aplicações e serviços). Assim, as performances musicais robóticas podem ser classificadas como atividades de loMusT.

#### 4. A Plataforma RoboMus

Dentro do contexto de ubimus e loMusT, um trabalho que vem sendo desenvolvido é a plataforma RoboMus (Camporez et al, 2018). A plataforma (Figura 1) possui uma arquitetura aberta de software e hardware tendo como objetivo a construção de robôs musicais de baixo custo controlados por humanos por intermédio de interfaces de interação. Apesar do foco do trabalho encontrar-se no controle por seres humanos, não se descarta a possibilidade de embutir módulos semiautônomos e autônomos para que possam atuar em determinados momentos da performance, por exemplo em uma desejada pausa rápida do usuário, nesse caso o robô assumiria a performance. A iniciativa pressupõe o uso de um formato predefinido de mensagens musicais, de modo que os robôs possam ser controlados e colaborarem mutuamente durante uma performance musical. Como é desejável uma configuração de múltiplos robôs em uma performance musical em tempo real, estratégias de sincronização são aplicadas através de um servidor de macrossincronização<sup>i</sup> de mensagens musicais, assim como, algoritmos de microssincronização<sup>ii</sup> que são embarcados nos robôs.

A RoboMus pode ser dividida em três partes: a) os robôs (instrumentos musicais robóticos), que incluem dispositivos comuns de controle a todos os robôs compatíveis com a plataforma; b) interfaces de controle dos robôs, que incluem a definição de linguagem de programação musical, interfaces colaborativas, pervasivas/ubíquas, tangíveis, gestuais, biossinais, entre outras; c) Servidor de Sincronização de Mensagens Musicais (SSMM), que executa o escalonamento de mensagens, empregando técnicas de compensação de atrasos mecânicos (macrossincronização). Também nessa linha, são propostas estratégias para compensar falhas de comunicação de dados durante a performance musical de tal forma que a audiência não perceba tais falhas. Por exemplo, se um robô

perder a conexão com o servidor, um outro pode fazer uma ponte de conexão para ele. Na Figura 1, pode-se observar a arquitetura da plataforma RoboMus.

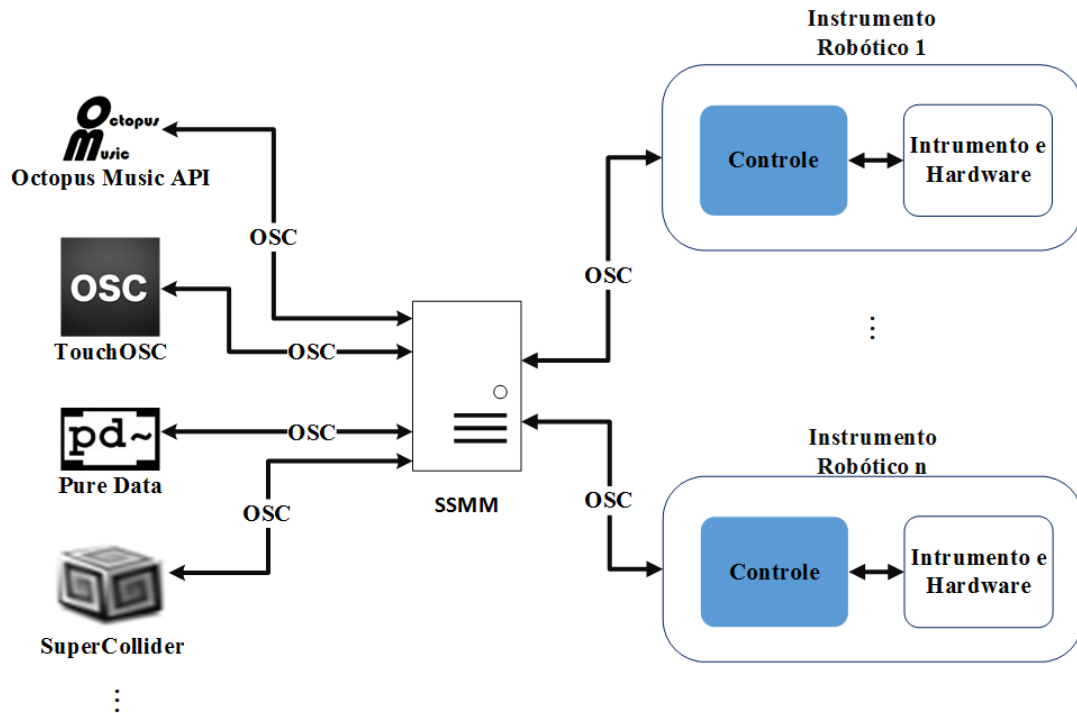


Figura 1. Arquitetura da Plataforma RoboMus

Na plataforma é empregado o protocolo de comunicação Open Sound Control (OSC) para definir os formatos de mensagens. Para a conexão entre o SSMM e os robôs, a plataforma segue o viés de que cada robô possui suas peculiaridades e limitações mecânicas. Isto posto, trabalha-se com uma espécie de apresentação (*handshake*), onde o robô envia suas informações e ações específicas de controle através de um padrão de comunicação predefinido. Vale ressaltar que ao se conectar em uma rede contendo uma sala de concerto, o robô pode encontrar um SSMM através de uma mensagem *broadcast* em uma porta de comunicação específica, para então, obter informações do servidor. No entanto, algumas mensagens básicas são definidas e todo robô RoboMus deve implementá-las, por exemplo, a mensagem *playNote* que contém os parâmetros: momento de execução, nota, duração e intensidade da nota. É possível que certos robôs não consigam executar tal ação com todos os detalhes especificados, assim, o mesmo deve contornar a situação para melhor executar a ordem. Além disso, padrões de mensagens para conexão das interfaces de controle com o SSMM são definidos para que as interfaces tenham conhecimento dos instrumentos e suas mensagens específicas. Assim sendo, ao se conectar a uma rede, a interface pode enviar uma mensagem *broadcast* de *handshake* para encontrar um SSMM.

#### 4.1. Robôs RoboMus

A plataforma RoboMus também apresenta dois protótipos robóticos visando o baixo custo e a reutilização de componentes, estes são: o *lap steel guitar* robótico e o bongo robótico (BongoBot). Além desses, neste artigo são descritos outros dois trabalhos envolvendo a robotização de uma flauta de pão e de um violino.

#### 4.1.1. Lap Steel Guitar Robótico

Os autores Camporez et al (2018) descrevem um protótipo robótico de uma *lap steel guitar* com 9 cordas em afinações diferenciadas do padrão (G4, F4, E4, C4, A3, F3, D3, A2 e D2). Para representar as ações de mão direita, no caso de um usuário destro, foram utilizados 18 servomotores, sendo 9 para excitação de cordas e 9 para abafamento das mesmas. Para o dispositivo de mão esquerda, foi utilizado um motor de passo e uma barra de *slide* presa em uma estrutura deslizante que se movimenta por intermédio do motor. Todos esses dispositivos são controlados por um Arduino e um Raspberry PI, podendo o Raspberry ser substituído por um *smartphone* Android (visando a reutilização de dispositivos antigos que costumam ficar abandonados em gavetas, ou acabam virando lixo eletrônico, por estarem ultrapassados tecnologicamente). A Figura 2 ilustra a arquitetura do robô. Resultados em vídeo desse robô podem ser encontrados em (NESCoM, 2018a).

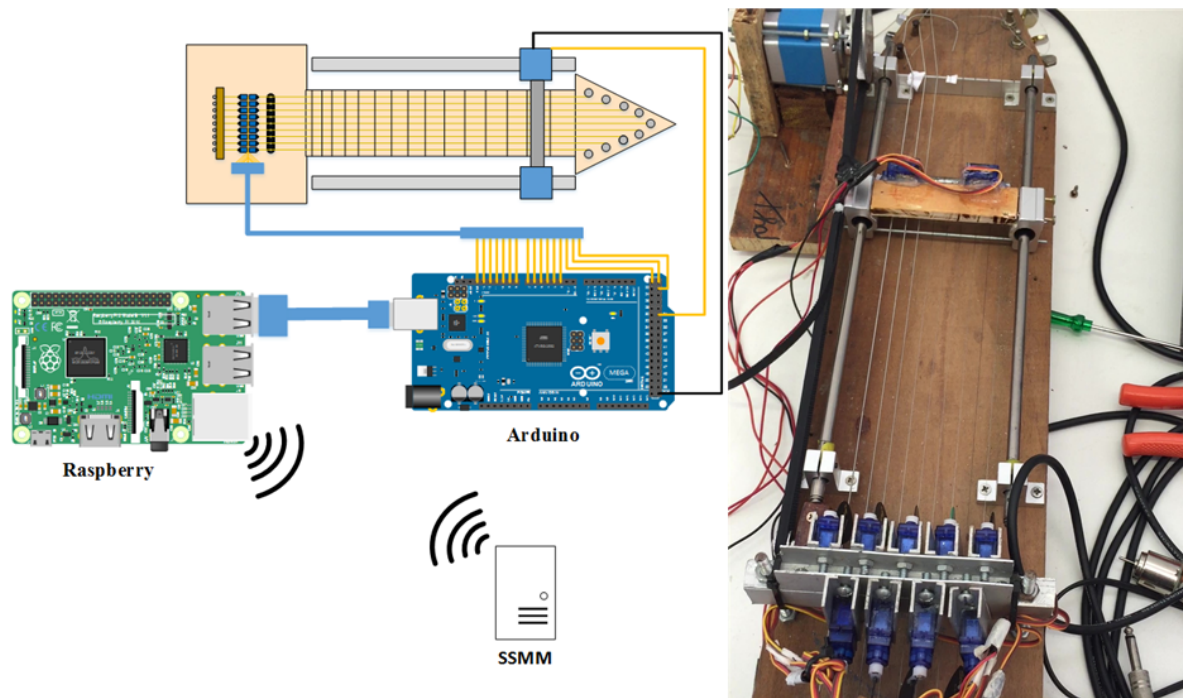


Figura 2. Lap Steel Guitar Robótico.

#### 4.1.2. BongoBot

Este protótipo foi desenvolvido para os primeiros testes de comunicação da plataforma. Chamado de BongoBot, e utilizando um bongô, o dispositivo foi criado com um hardware que utiliza dois servomotores, duas baquetas para xilofone, um *smartphone* Android e um Arduino Uno (Figura 3). Vídeos demonstrativos podem ser encontrados em (NESCoM, 2018b).

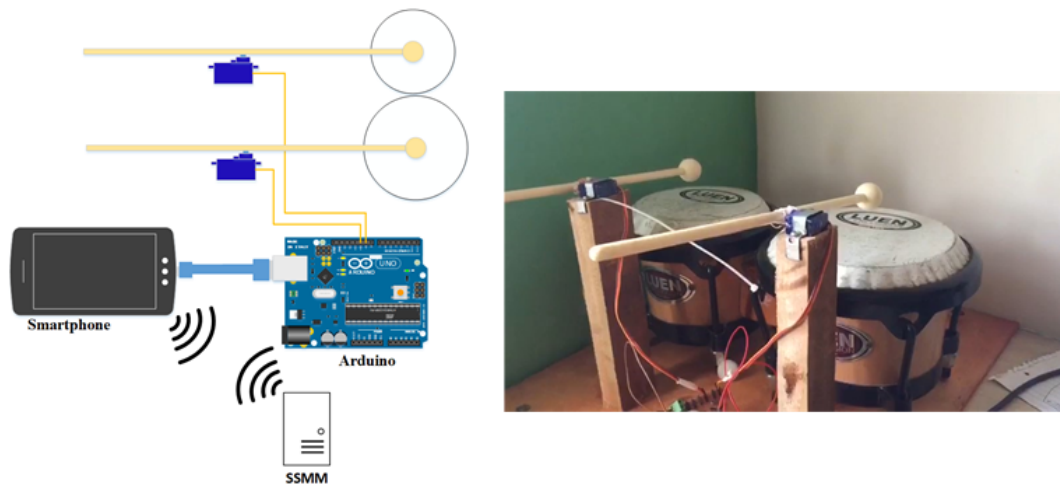


Figura 3. Bongo Robótico (BongoBot).

#### 4.1.3. Flauta de Pã

Ainda com viés de projetos de baixo custo, apresenta-se um protótipo de uma flauta de pã robótica. A flauta foi construída com partes de cano pvc, contendo 11 notas no total. Para os aparatos tecnológicos, utilizou-se um motor de passo, um *driver* para o motor, um compressor de ar e um Arduino. A flauta foi fixada em um pedaço de aro e o motor de passo foi conectado a estrutura para proporcionar rotação. Com a mangueira de ar proveniente do compressor fixada, a rotação proporcionou a mudança de nota na flauta. A Figura 4 ilustra a estrutura do protótipo.

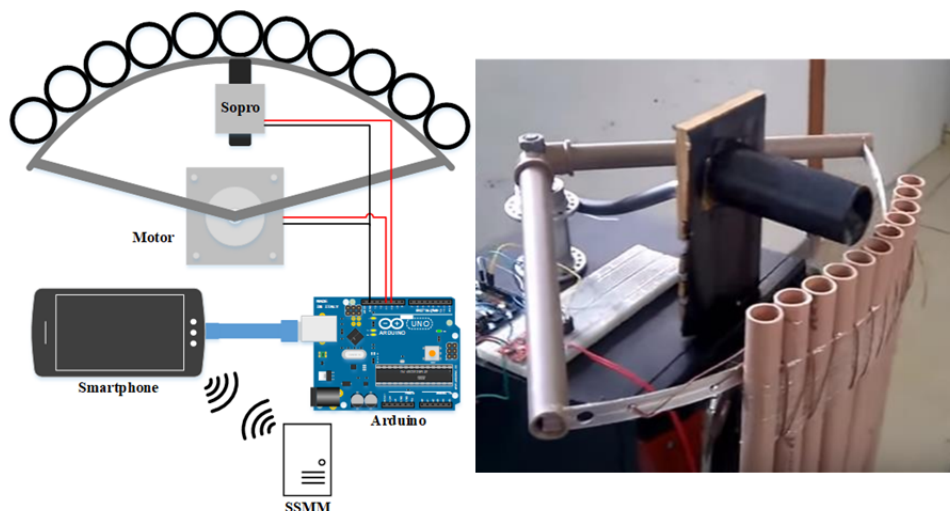


Figura 4. Flauta de Pã Robótica.

#### 4.1.4. Violino Robótico

Um protótipo de violino robótico também foi implementado para interação com a plataforma RoboMus. O protótipo se dividiu em três frentes de trabalho: (a) movimento do arco; (b) giro do violino e (c) pressionamento de corda. Na frente de trabalho (a) visou-se emular o movimento do arco

para produção sonora, assim o arco foi preso em um dispositivo para movimentá-lo horizontalmente. Em (b) foi desenvolvido o dispositivo para rotacionar o violino com o objetivo de alternar a corda em que o arco estava tocando, a estrutura faz uso de um motor de passo para tal fim. Já em (c) foi construída uma estrutura para pressionar as cordas, onde utilizou-se dois servomotores em posições fixas. A Figura 5 demonstra arquitetura supracitada.

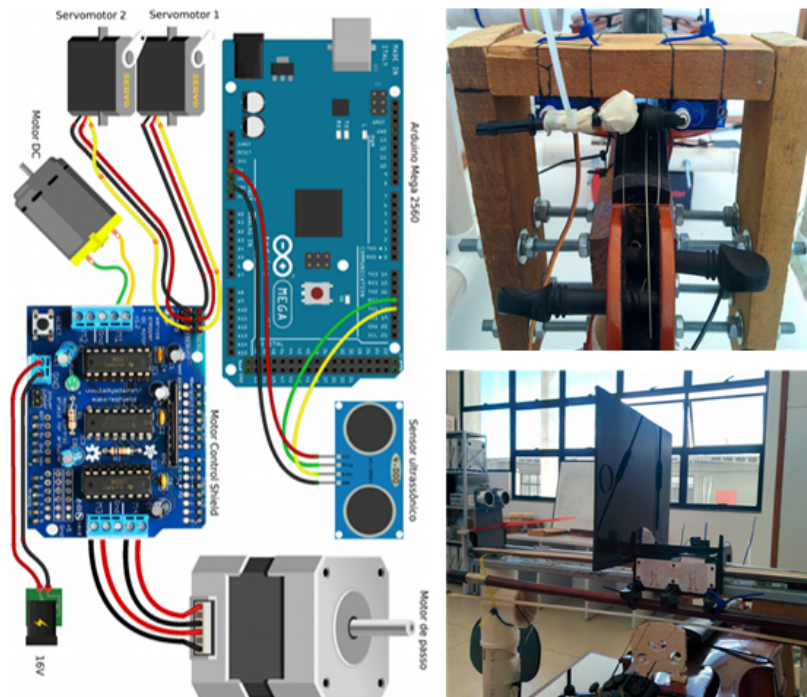


Figura 5. Violino Robótico.

## 4.2. Interfaces de Interação Humano-Robô (RHI)

As interfaces de interação entre seres humanos e robôs vem sendo bastante exploradas, inclusive no âmbito de instrumentos musicais robóticos, como por exemplo a interface gestual para controle de um robô flautista (Petersen, Solis, & Takanishi, 2008). Contudo, em geral, os músicos de uma banda trocam informações principalmente através da percepção visual e acústica. Sendo assim para uma performance mais realista pode-se implementar nos robôs mecanismos que sejam capazes de recriar tais interações. O marimba robótico *Shimon* (Hoffman & Weinberg, 2010) consegue interagir em tempo real com músicos, tanto no acompanhamento acústico quanto na interação gestual, uma vez que possui características físicas análogas as dos seres humanos. Outro exemplo, é o robô *Shimi* (Bretan & Weinberg, 2014) que possui 5 graus de liberdade para a interação gestual, caixas acústicas para reprodução sonora e utiliza um *smartphone* como interface gráfica, além de utilizar o microfone e a câmera para a entrada de dados visando interação.

Como o controle dos robôs por seres humanos é uma das premissas da plataforma, idealiza-se interfaces de controle desenvolvidas por intermédio tanto de uma linguagem de programação musical, de uma API, quanto de um dispositivo controlador - por exemplo: *WiiMote*, *smartphone*, instrumentos inteligentes e etc. Assim, há um alinhamento aos sistemas de ubimus quando suportam

múltiplas formas de interação e múltiplas fontes sonoras (geradas pelos robôs) de forma integrada (Keller et al, 2014).

A plataforma RoboMus também visa o desenvolvimento de interfaces assistivas para a inclusão de pessoas portadoras de deficiência motora. Assim, ainda em processo de desenvolvimento, tem-se uma interface chamada Olhar Musical (Camporez, Neto, Costalonga, & Rocha, 2018) que faz uso de um dispositivo para rastreamento de olho (*eye tracker*). Todo controle da interface, e, por conseguinte, dos robôs, é feito por intermédio do movimento dos olhos. Em primeiro momento a interface foi desenvolvida para execução de amostras de áudio previamente configuradas na interface (Figura 6), visando o controle de robôs como trabalhos futuros. Toda amostra de áudio é executada pelo software Ableton Live através de mensagens de controle enviadas pela interface. Na Figura 6 pode-se observar as células centrais que representam uma amostra de áudio cada, assim, o usuário pode selecionar uma célula central e o botão *play*, para iniciar a execução, ou *pause*, para encerrar uma execução. Os botões *play* e *pause* do lado esquerdo da Figura 6 tem objetivo de fornecer, ao usuário, a possibilidade de conhecimento prévio das amostras através de um canal de áudio separado da performance. Já os botões do lado direito executam a amostra diretamente na performance.



**Figura 6.** Interface Assistiva Olhar Musical.

#### 4.3. Sincronismos dos Robôs Musicais

Em performances musicais a sincronia entre os participantes é essencial, no contexto de performances musicais robóticas não é diferente. Os humanos possuem atrasos mecânicos e os compensam iniciando a execução do movimento momentos antes da desejada resposta sonora do instrumento. Os robôs, por possuírem componentes mecânicos, também apresentam tais atrasos. Nesta seção é descrito sobre algumas estratégias de macrossincronização promovidas pelo SSMM para compensá-los.

A plataforma RoboMus propõe enviar o tempo de execução da mensagem utilizando tempo absoluto, o *Time Tag*, presente no protocolo OSC, que contém o tempo em segundos desde a meia-noite de 1 de janeiro de 1970, com precisão de picossegundos. Contudo, há uma necessidade de sincronizar os relógios dos robôs para que todos possuam a mesma referência. Assim, para

fornecer essa sincronia, propõe-se a implementação do protocolo PTP (*Precision Time Protocol*), definido no padrão IEEE 1588 (IEEE, 2008), que fornece a hora com precisão através de uma rede. Nesse caso, o servidor SSMM também funciona como um servidor PTP e os robôs como clientes PTP. Estudos com o protocolo PTP mostram um erro no relógio de 2,8  $\mu$ s (Kannisto, Vanhatupa, Hännikäinen, & Hämäläinen, 2004) e um erro médio de 7,35  $\mu$ s (Cooklev, Eidson, & Pakdaman, 2007) para WLAN (*Wireless Local Area Network*), valores plausíveis frente a percepção sonora humana que é capaz de distinguir diferenças temporais entre dois eventos acústicos sucessivos em até 20 milissegundos (Pisoni, 1977).

Cada robô da plataforma possui um perfil de atraso próprio devido ao seu projeto estrutural. Assim, é crucial que o SSMM tenha conhecimento desses perfis para a aplicação de técnicas de sincronização sobre os robôs. Para tal, é proposto um treinamento prévio, ou seja, o servidor envia a um robô uma mensagem contendo uma ação e ele retorna uma mensagem contendo o tempo de atraso que gastou para executar tal ação. Esse tempo é contabilizado a partir do momento que o robô envia os sinais elétricos para seus atuadores até o momento em que é gerada a resposta sonora. O formato da mensagem contendo o atraso segue o conteúdo descrito na **Tabela 1**, onde o endereço OSC é a junção entre o endereço do servidor, a palavra *delay* e o endereço do instrumento remetente. Já o primeiro parâmetro é o identificador da mensagem (o mesmo usado na mensagem enviada pelo servidor) e o segundo parâmetro é o atraso mecânico.

**Tabela 1.** Formato da Mensagem para Retorno do Atraso Mecânico.

|                    |                                                             |
|--------------------|-------------------------------------------------------------|
| <b>Endereço</b>    | /Endereço OSC do servidor/delay/Endereço OSC do instrumento |
| <b>Parâmetro 1</b> | Identificador                                               |
| <b>Parâmetro 2</b> | Atraso mecânico                                             |

Com os atrasos e as mensagens, o servidor pode então levantar o perfil de atraso para cada robô utilizando redes neurais. Nesse caso, o SSMM envia centenas de mensagens para cada robô e levanta seu perfil, esse processo demanda certo tempo, tanto para envio e execução das ações quanto para treinamento da rede para cada robô. No entanto, todo processo é feito nos bastidores e não no momento da performance.

Através do *handshake*, o SSMM tem conhecimento de várias informações de um robô, entre essas destaca-se a informação de ações específicas, que tem o objetivo de transmitir ao SSMM todas ações que o robô pode executar e os parâmetros necessários para o controle do mesmo. Contudo, para que o SSMM crie as mensagens para treinamento é necessário ter o conhecimento dos conjuntos de valores válidos para cada parâmetro, para que desse modo o SSMM possa montar mensagens válidas e facilitar a generalização e o treinamento da rede neural. O protocolo de ação específica do RoboMus proposto por Camporez et al (2018) não prevê esses conjuntos, como pode ser visto no exemplo a seguir:

```
</ação1;  parâmetro11_tipo;  ...  ;  parâmetro1N_tipo>  ...  </açãoN;
parâmetroN1_tipo;  ...  ;  parâmetroNN_tipo>.
```

O formato acima identifica cada ação entre "<>" e, dentro cada ação, as informações são separadas por ";", onde o primeiro componente representa o endereço OSC da ação e os demais são os parâmetros que contém o nome e tipo do parâmetro separado por "\_".

Assim, neste artigo é proposto um novo formato para a informação de ações específicas, onde inclui-se os conjuntos válidos para cada parâmetro de cada ação. Ao final do formato anterior é adicionado o novo padrão onde os valores válidos são representados entre parêntesis e os conjuntos separados por ";", exceto o primeiro termo que é o nome da ação. Abaixo segue o novo modelo do padrão:

```
</ação1; parâmetro11_tipo; parâmetro12_tipo; ... ; parâmetro1N_tipo> ...  
</açãoN; parâmetroN1_tipo; ... ; parâmetroNN_tipo>(/ação1; conjunto11;  
conjunto12; ... ; conjuntoN) ... (/açãoN; conjunto1N; conjunto2N; ... ;  
conjuntoNN).
```

Cada conjunto pode ser representado de duas formas. Uma que determina o intervalo com limite superior e inferior, como por exemplo,  $]a,b]$  onde  $a$  é o limite inferior, mas não pertence ao conjunto, representado por  $]a$ , e  $b$  é o limite superior que pertence ao intervalo, representado por  $b]$ , seguindo o padrão matemático de intervalos. A segunda forma é a passagem literal dos possíveis valores, nesse caso os valores são contidos entre chaves ( $\{\}$ ) e separados por vírgula, por exemplo,  $\{p1, p2, p3, \dots, pn\}$ . Segue um exemplo:

```
</playNote; string_i; fret_i></playString;string_i>(/playNote; {1, 2, 3,  
4}; [1, 12])(/playString; [1, 6]).
```

No exemplo anterior o instrumento possui duas ações: `playNote` e `playString`. A ação `playNote` possui dois parâmetros `string_i` e `fret_i`, pode-se observar pelo caractere `i` que ambos são do tipo inteiro, seus conjuntos são  $\{1,2,3,4\}$  e  $[1,12]$ , respectivamente. Já a ação `playString` possui apenas um parâmetro `string_i` do tipo inteiro, que possui o intervalo  $[1,6]$ . Quando um parâmetro não possui intervalo, o valor deve ficar vazio, como no exemplo abaixo, onde `duration_i` não tem intervalo,

```
</playNote; string_i; duration_i; velocity_i>(/playNote;{1, 2, 3, 4}; ;[1,  
12]).
```

Para facilitar a leitura dos possíveis valores dos parâmetros pelo SSMM e, com isso, facilitar o treinamento da rede, mais um tipo de parâmetro é definido para a plataforma RoboMus: o tipo nota com identificador  $n$ . Esse novo tipo segue o padrão universal para notas utilizando caracteres ( $C =$  Dó,  $D =$  Ré,  $E =$  Mi, ...). O padrão segue símbolo da nota + acidente + oitava, por exemplo  $A\#4$ . Segue um exemplo de uma ação específica utilizando esse modelo:

```
</playNote; note_n; duration_i>(/playNote; {C4, C#4, D4, E4}; ]0, 1000]).
```

## 5. Conclusão

Esse artigo apresentou os principais conceitos de IoMusT (*Internet of Musical Things*) inseridos no contexto da música ubíqua (ubimus), bem como uma aplicação prática através da plataforma RoboMus com o objetivo de promover performances musicais robóticas em tempo real, controladas por seres humanos, sem descartar o uso de módulos semiautônomos ou autônomos para auxiliar o usuário. Assim, descreve-se a construção de protótipos de robôs musicais; a interface de controle dos robôs; uma proposta para sincronização entre robôs usando o protocolo PTP (*Precision Time Protocol*) para sincronização de relógios, redes neurais para treinamento e compensação dos perfis de atraso ocasionados pelos atuadores mecânicos dos robôs.

As modificações nos formatos das mensagens para propiciar o uso de redes neurais para aprendizado dos atrasos mecânicos traz maior robustez para a plataforma. Acredita-se que os desgastes físicos dos componentes mecânicos possam interferir no atraso ao longo do tempo, nesse caso, a rede pode ser novamente treinada para aprender os novos atrasos. Além disso, no momento da performance podem ocorrer erros devidos ao aquecimento de componentes mecânicos. Isso pode ser previsto pela rede.

Como trabalhos futuros visa-se o uso de inteligência artificial em módulos semiautônomos, implementados nas interfaces de controle para auxiliar os usuário, principalmente os não músicos, a tomar decisões, por exemplo apresentando possibilidades de próximos eventos compatíveis; módulos autônomos também são considerados para possibilitar uma espécie de piloto automático, visando auxiliar o usuário em casos de ausência de curta duração no meio da performance, nesse caso o robô continuaria atuando na performance de acordo com o contexto. Em relação a sincronização entende-se a necessidade de levantar dados técnicos experimentais, através da medição dos atrasos por intermédio de sensores sonoros e também uma avaliação por grupos de usuários, para assim comprovar tanto a sincronia quanto a eficácia da proposta.

## Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## Referências

- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/https://doi.org/10.1016/j.comnet.2010.05.010>
- Boley, H., & Chang, E. (2007). Digital ecosystems: Principles and semantics. *Digital EcoSystems and Technologies Conference*, 398–403.
- Bretan, M., & Weinberg, G. (2014). Chronicles of a Robotic Musical Companion. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 315–318. Retrieved from [http://www.nime.org/proceedings/2014/nime2014\\_303.pdf](http://www.nime.org/proceedings/2014/nime2014_303.pdf)
- Bretan, M., & Weinberg, G. I. L. (2016). A Survey of Robotic Musicianship. *Communications of the ACM*, 59(5), 100–109. <https://doi.org/10.1145/2818994>
- Camporez, H. A. F., Mota, T. S. R., Astorga, E. M. V., Rocha, H. R. O., & Costalonga, L. L. (2018). RoboMus: Uma Plataforma para Performances Musicais Robóticas. In D. Keller & M. H. De Lima (Eds.), *Aplicações em Música Ubíqua* (1st ed., pp. 58–93). São Paulo, Brasil: ANPPOM.

- Camporez, H. A. F., Neto, A. F., Costalonga, L. L., & Rocha, H. R. de O. (2018). Interface Computacional para Controle Musical Utilizando os Movimentos dos Olhos. *Revista Vórtex*, 6(2), 1–17. Retrieved from [http://vortex.unespar.edu.br/camporez\\_et\\_al\\_v6\\_n2.pdf](http://vortex.unespar.edu.br/camporez_et_al_v6_n2.pdf)
- Cooklev, T., Eidson, J. C., & Pakdaman, A. (2007). An Implementation of IEEE 1588 Over IEEE 802.11b for Synchronization of Wireless Local Area Network Nodes. *IEEE Transactions on Instrumentation and Measurement*, 56(5), 1632–1639. <https://doi.org/10.1109/TIM.2007.903640>
- Delicato, F. C., Pires, P. F., Batista, T., Cavalcante, E., Costa, B., & Barros, T. (2013). Towards an IoT Ecosystem. *Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems*, 25–28. <https://doi.org/10.1145/2489850.2489855>
- Hoffman, G., & Weinberg, G. (2010). Shimon: An Interactive Improvisational Robotic Marimba Player. *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, 3097–3102. <https://doi.org/10.1145/1753846.1753925>
- IEEE. (2008). IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, 1–300. <https://doi.org/10.1109/IEEESTD.2008.4579760>
- Kannisto, J., Vanhatupa, T., Hännikäinen, M., & Hämäläinen, T. D. (2004). Precision Time Protocol Prototype on Wireless LAN. In J. N. de Souza, P. Dini, & P. Lorenz (Eds.), *Telecommunications and Networking - ICT 2004* (pp. 1236–1245). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Kapur, A. (2005). A History of robotic Musical Instruments. *ICMC*.
- Keller, D., Flores, L. V., Pimenta, M. S., Capasso, A., & Tinajero, P. (2011). Convergent Trends Toward Ubiquitous Music. *Journal of New Music Research*, 40(3), 265–276. <https://doi.org/10.1080/09298215.2011.594514>
- Keller, D., & Lazzarini, V. (2017). Ecologically Grounded Creative Practices in Ubiquitous Music. *Organised Sound*, 22(1), 61–72. <https://doi.org/10.1017/S1355771816000340>
- Keller, D., Lazzarini, V., & Pimenta, M. S. (Eds.). (2014). *Ubiquitous Music*. Springer Publishing Company, Incorporated.
- NESCoM. (2018a). RoboMus - Lapsteel guitar. Retrieved February 20, 2020, from [goo.gl/KHrw4k](http://goo.gl/KHrw4k)
- NESCoM. (2018b). RoboMus - Teste BongoBot. Retrieved February 20, 2020, from [goo.gl/V6nvSn](http://goo.gl/V6nvSn)
- Petersen, K., Solis, J., & Takanishi, A. (2008). Toward enabling a natural interaction between human musicians and musical performance robots: Implementation of a real-time gestural interface. *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*, 340–345. <https://doi.org/10.1109/ROMAN.2008.4600689>
- Pisoni, D. B. (1977). Identification and discrimination of the relative onset time of two component tones: implications for voicing perception in stops. *The Journal of the Acoustical Society of America*, 61(May 2013), 1352–1361. <https://doi.org/10.1121/1.381409>
- Singer, E., Feddersen, J., Redmon, C., & Bowen, B. (2004). LEMUR's Musical Robots. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 181–184. <https://doi.org/10.1145/1027527.1027569>
- Turchet, L., Fischione, C., Essl, G., Keller, D., & Barthet, M. (2018). Internet of Musical Things: Vision and Challenges. *IEEE Access*, 6, 61994–62017. <https://doi.org/10.1109/ACCESS.2018.2872625>

---

<sup>i</sup> Técnicas de sincronização aplicada aos robôs por um agente externo (SSMM), fornecendo uma sincronia básica.

<sup>ii</sup> Técnicas de sincronização aplicada pelos próprios robôs para ajuste fino da sincronia com os demais.