

# MOVIMENTAÇÃO AUTÔNOMA DE ROBÔ HUMANOIDE COM CÂMERA EMBARCADA

Raphael Ballet <sup>1</sup>; Alessandra Dutra Coelho <sup>2</sup>; Wânderson de Oliveira Assis <sup>2</sup>

<sup>1</sup> Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM-CEUN-IMT);

<sup>2</sup> Professor(a) da Escola de Engenharia Mauá (EEM-CEUN-IMT).

**Resumo.** *O objetivo deste trabalho é o desenvolvimento do controle de movimentação autônoma de um robô humanoide através de uma câmera embarcada e sensores inerciais. O projeto é desenvolvido na integração da estrutura hierárquica de controle desenvolvida anteriormente, de novas ferramentas de sensoramento e comunicação sem fio e da criação de algoritmos eficientes de controle e de comunicação, assim possibilitando ao robô obter mais informações sobre o ambiente em que ele se encontra e poder realizar decisões corretas para sua correta movimentação. Estas decisões são baseadas em sensores que assemelham ao sistema de sensoramento de um ser humano, o que é o objetivo fundamental de um robô humanoide, e a principal fonte de informação é proveniente do processamento de imagens da câmera embarcada, o que traz a proximidade com o sistema de visão humano.*

## Introdução

A utilização de câmeras embarcadas em robôs autônomos vem se tornando uma prática comum em pesquisas ou mesmo na indústria atual. É muito comum ver universidades do mundo inteiro e mesmo empresas públicas ou privadas investindo em tecnologias de automação através de sistema de processamento de imagens com câmeras embarcadas e outros sensores. Uma aplicação então utilizada é o rastreamento de superfícies, como câmeras acopladas em aviões, helicópteros autônomos ou mesmo em carros autônomos. Outra importante aplicação é o uso em veículos submarinos, como o feito por Crétual e Chaumette (2000), ou mesmo na medicina com cirurgias não presenciais (Palep, 2009).

Apenas uma pequena parcela dos robôs é composta de robôs considerados humanoides, mas é de grande importância seu estudo. Com este tipo de robô é possível a criação de algoritmos complexos de controle devido a seu elevado número de graus de liberdade, os quais simulam, de forma simplificada, as articulações humanas. Através do estudo da movimentação de um robô humanoide, podem-se desenvolver conhecimentos sobre a movimentação humana (Asada *et al.*, 2011) ou, até mesmo, em desenvolver próteses móveis controladas pelo cérebro (Cui *et al.*, 2006)

Visto suas utilidades, o enfoque será no uso de robôs autônomos, ou seja, robôs que não tem nenhuma interferência humana. Eles devem ser capazes de “perceber” o ambiente e atuar conforme o objetivo a ele designado. Para isso, estes robôs devem obter informações suficientes do ambiente em que ele se encontra e um controlador capaz de processar todos estes dados para conseguir realizar a decisão correta de movimentação.

Este projeto é a continuação e aperfeiçoamento do trabalho desenvolvido no artigo (Ballet, 2012), na qual houve a introdução de um sistema hierárquico de tomada de decisões e a criação de um sistema pan-tilt para movimentação da câmera embarcada. Através do trabalho anterior, foi visto a dificuldade da realização do controle de um robô humanoide, já que este apresenta apenas dois apoios, sendo que apenas um deles está em contato com o solo no movimento de caminhada. A realização do controle exato de movimentação do robô é de extrema complexidade e envolve teorias de mecânica inversa para cada servo-motor, o que não será tema deste trabalho. Para isso, preferiu-se adotar os movimentos já programados de fábrica na placa

MRC-3024 que permitem o robô realizar movimentos simples de caminhada e de posicionamento. Estes movimentos foram aperfeiçoados, obtendo assim um movimento mais controlado em algumas situações, já que, com a adição de novos equipamentos, o centro de massa do robô foi deslocado, deixando o robô mais instável.

O trabalho apresenta o objetivo de obter um controle eficiente da movimentação do robô humanoide através de uma câmera embarcada, assim como promover a integração da estrutura hierárquica de tomada de decisões com sensores inerciais e com um dispositivo de comunicação sem fio para obter informações em tempo real do robô enquanto ele realiza a movimentação e permitir o robô se comunicar com outros robôs, o que permite os robôs realizarem estratégias de movimentação em conjunto. Para realizar a telemetria, foi criado um programa no ambiente *Processing®*, um ambiente de programação de linguagem gráfica, que permite a recepção de dados em tempo real. Com este programa, torna-se possível tornar o robô autônomo ou que funcione através do computador, assim como receber informações sobre o objeto encontrado na câmera e qual decisão tomada por ele enquanto ele realiza a ação.

A forma de controle desenvolvida no artigo (Ballet, 2012) demonstrou ser a forma mais eficiente de controle com relação a seu custo e facilidade de programação. Ela é baseada em três níveis de controle, sendo que há um nível de comando central que realiza a comunicação com os outros dois. O primeiro nível é a partir da CMUCAM4, uma placa que contém um sensor CMOS de câmera. Essa placa tem a função de capturar a imagem, realizar o processamento de imagens e mandar para o segundo e mais importante nível. O segundo nível é composto por uma placa *Arduino*, um hardware livre com o qual há diversos projetos e bibliotecas presentes, o que torna rápido e fácil o trabalho de programação. O *Arduino* será o comando central do robô. Através dos comandos indicados pelo *Arduino*, o terceiro nível é acionado e ele corresponde ao sistema já existente no robô, de fábrica. A placa MRC-3024, presente no robô, recebe a ação desejada pelo *Arduino* e realiza os movimentos existentes.

## **Material e Métodos**

### Materiais utilizados

O robô humanoide utilizado é o Robonova-1 (figura 1) produzido pela *HiTec*. O robô contém 16 servomotores do tipo HSR-8498 e uma placa de controle MRC-3024 (figura 2). A placa MRC-3024 contém um microcontrolador do tipo ATMEL ATMEGA 128 e é programada por uma interface em linguagem específica do Robonova-1, o *RoboBasic*, que é uma linguagem simples e com poucos recursos. Através desta limitação de programação, houve a necessidade de obter alternativas para um melhor controle de ações e inteligência.



Figura1 -Robonova-1.



Figura 2- MRC-3024.

O sistema de visão utilizado no projeto é composto por uma câmera embarcada CMUCam4 (figura 3) e foi implantado um sistema de movimentação com 2 graus de liberdade, sistema *pan-tilt*. Este sistema é utilizado para movimentação nos três eixos, o que permite a liberdade de movimentação necessária para a ação desejada. A CMUCam4 apresenta um módulo de sensor de câmera CMOS 9665 com um processador Parallax P8X32A. A placa é *open-source* e pode ser reprogramada de diversas maneiras. A maneira que será utilizada para esse projeto será através da biblioteca existente da CMUCam4 para o Arduino. A placa CMUCam4 será a responsável por todo o processamento de imagens e cálculo dos parâmetros da imagem observada. Para conseguir reconhecer um objeto, é necessário dar os parâmetros RGB, que são os valores de vermelho, verde e azul, respectivamente, que são específicos do objeto que se deseja procurar na imagem. Esses parâmetros são enviados a partir de um intervalo de valores, para garantir que a câmera consiga visualizar toda a bola mesmo com níveis diferentes de luminosidade. Após essa análise de cores, a CMUCam4 envia pacotes de dados (tipo T) contendo diversos parâmetros como:

-mx = valor médio (centro geométrico) horizontal;

-my = valor médio (centro geométrico) vertical;

-pixels = A porcentagem de pixels encontrados com os parâmetros desejados na imagem;

-confidence = A densidade de pixels reunidos em uma parte da imagem (usado para determinar se há uma região de certa cor, como uma bola, por exemplo).

Essas informações são enviadas continuamente para o Arduino e é a partir delas que é feita toda a lógica desejada.



Figura 3-CMUCam4.

O sistema de controle foi definido a partir da necessidade de prover maior nível de inteligência ao robô, podendo haver uma melhor programação e controle das atividades necessárias. O robô apresenta uma placa MRC-3024 que apresenta um microcontrolador ATMEL ATMEGA 128. Esta placa apresenta diversas limitações de programação, o que não permite a

inclusão da lógica de controle nela. Para isso, foi desenvolvido um sistema descentralizado de controle, baseado em uma estrutura hierárquica de tomada de decisão. Nesta estrutura criada, foi introduzido um microcontrolador Arduino, que realiza o controle central de decisões.

A placa Arduino utilizada é modelo Arduino Nano (figura 4) e apresenta as seguintes características:

-Microcontrolador	-Atmel ATmega328
-Voltagem de operação (nível logico)	-5 V
-Pinos digitais I/O	-14 (dos quais 6 podem ser saídas PWM)
-Pinos de entrada analógica	-8
-SRAM	-1 KB
-EEPROM	-512 bytes
-Velocidade de Clock	-16 MHz
-Dimensões	-0.73" x 1.70"

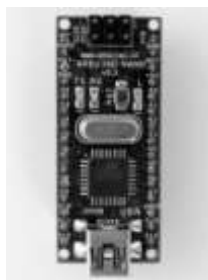


Figura 4- Arduino Nano.

O Arduino Nano é programado através de um cabo Mini-B USB, que também pode ser usado para comunicação serial e portas Rx/Tx, que são usadas para comunicação serial TTL. Há também a possibilidade de comunicação SPI e I<sup>2</sup>C. O Arduino possui diversas bibliotecas disponíveis para diversas situações, o que torna mais rápida a criação de códigos mais complexos. Por ser um hardware *open-source*, ou seja, todo seu esquema eletrônico é aberto ao público, existe uma grande variedade de cópias e *Shields* no mercado. Os *Shields* são placas com elementos adicionais que podem ser montadas em cima de outra placa Arduino, aumentando a quantidade de funções presentes, como relés, controladores de motor, LCD, GPS e até mesmo a própria CMUCam4.

Os sensores adicionais usados no robô são os acelerômetros e giroscópios. Os acelerômetros medem as acelerações em um eixo específico, o que permite obter informações sobre impactos, forças ou até a queda livre de um objeto. Através dele podemos analisar quais forças estão agindo sobre o robô e obter um melhor equilíbrio quando houver um movimento não desejado. O sensor de acelerômetro utilizado foi o sensor analógico ADXL320 de dois eixos. Ele gera uma saída entre 0,3 e 2,5 V. Este sensor foi utilizado para verificar a queda do robô.

Os giroscópios são muito importantes, principalmente em sistemas de navegação de aeronaves, navios, piloto automático, voos espaciais e etc. Ele se baseia em dois conceitos importantes da física que são a inércia e a precessão. O giroscópio é constituído basicamente de

um rotor suspenso com um eixo em rotação. Esse rotor mantém uma posição fixa mesmo após ser forçada em uma determinada direção não paralela à roda. Através desse sistema pode se observar a direção do objeto em relação ao rotor, o que permite a correta orientação do corpo. O sensor utilizado como giroscópio neste projeto foi o sensor da Hitec PG-03 de um eixo. Este sensor é utilizado para compensar as possíveis variações do corpo do robô enquanto ele esta se movimentando.

Em conjunto, o giroscópio e o acelerômetro constituem a base do IMU (*Inercial Measurement Unit*), que são equipamentos eletrônicos capazes de medir a aceleração, orientação e direção do movimento. Podem ser integrados ainda GPS, magnetômetros e até tubos de Pitot, para um completo controle de movimento no espaço.

Além dos sensores, foi adicionado um importante sistema de análise e pesquisa em tempo real, que é o uso de dispositivos de telemetria por tecnologia de rede sem fio, ou *wireless*. Neste trabalho, foi introduzido um dispositivo de tecnologia Zigbee®, através de um modem Xbee®, da empresa Digi International. Através deste dispositivo, é possível obter informações em tempo real do robô e analisar as possíveis causas de erro ou se a estratégia de movimentação está correta. Além disso, foi possível passar parâmetros da imagem através deste modem e colocar estes parâmetros em um programa gráfico, o que permite saber exatamente a visualização do objeto visto pela câmera. Não seria possível a passagem de todos os valores de pixels através do modem Xbee, mas através dos parâmetros enviados pela placa CMUCam4, podemos reproduzir a imagem vista neste programa desenvolvido em ambiente Processing®. Este modem pode ser visto na figura 5.



Figura 5 – Modem Xbee de comunicação wireless

Outro importante uso do dispositivo de comunicação sem fio Xbee é a possibilidade de criar redes de dispositivos intercomunicados. Este modem possui um sistema eficiente de codificação e canais, onde apenas o dispositivo configurado com uma determinada senha e um determinado canal que receberá as informações desejadas. Assim, pode-se criar uma rede, onde os robôs de diferentes senhas, mas em um mesmo canal, podem se intercomunicar e todos eles podem se comunicar com um dispositivo “mestre”, que pode ser o computador. Podendo, assim, criar uma equipe sincronizada de robôs autônomos.

### Projeto e testes

Como já dito, o projeto consiste em tornar possível um robô humanoide se orientar através de uma câmera embarcada e demais sensores e, através disso, ser possível realizar operações simples, como andar, dar um passo para o lado, chutar, levantar, sentar, virar e etc. O

controle será baseado em níveis de controle bem definidos, nos quais são necessárias comunicações bilaterais, de forma a tornar o sistema o mais robusto possível. A seguir está esquematizado o sistema de controle através da figura 6:

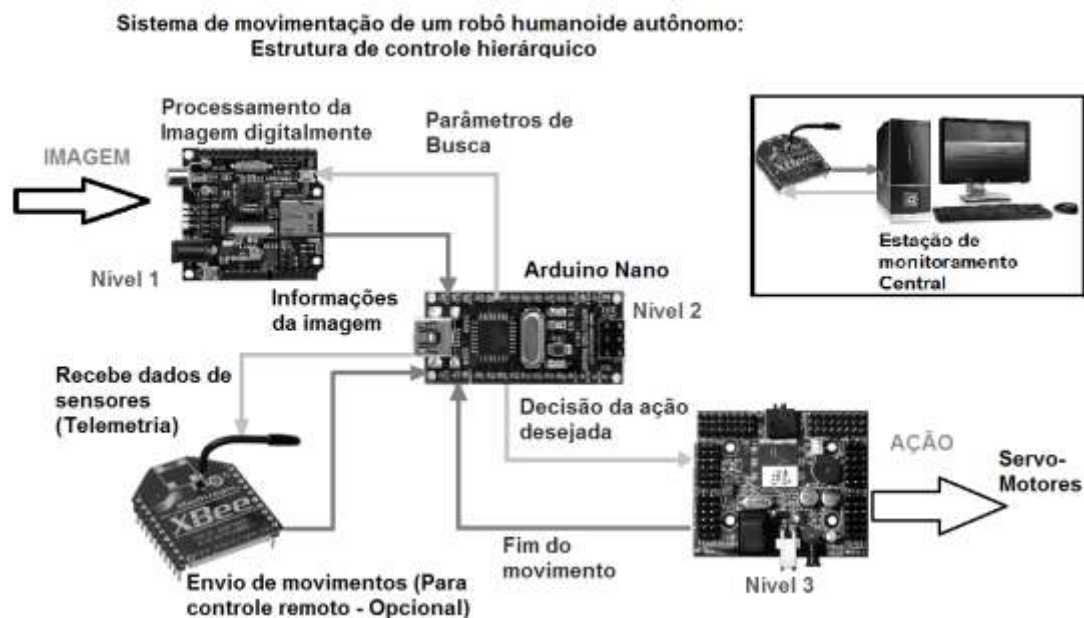


Figura 4 - Esquema de controle do robô.

O controle de posicionamento da câmera foi realizado através de um controle Proporcional-Derivativo, como visto no trabalho (Ballet, 2012). O controle interno do Arduino foi feito através do algoritmo simplificado a seguir:

1. Inicia a câmera com os valores RGB do objeto desejado;
2. Coloca todos os servo-motores nas posições iniciais;
3. Recebe os dados da câmera e analisa se houve objeto encontrado;
4. Recebe dados do computador e envia os dados da câmera para o computador;
5. Coloca os valores do centroide do objeto encontrado no controlador PD do Pan-Tilt;
6. Corrige os servo-motores para colocar o objeto no centro da câmera;
7. Se o servo-motor de controle horizontal estiver com mais de 30 graus de diferença com a posição frontal do robô, faça o robô girar o corpo;
8. Enquanto o objeto não estiver perto o suficiente do robô, faça o robô andar para frente;
9. Se robô estiver caído, levante;
10. Se não houver objeto visto, procure objeto através da movimentação do robô para os lados;
11. Quando a ação estiver decidida, espere o comando da placa MRC-3024 e então envie a decisão;
12. Enquanto não estiver perto do objeto, volte ao passo 3;

Através deste algoritmo, houve a possibilidade de decisão de forma simplificada. A decisão é feita através de controles hierárquicos de decisão. Cada decisão pode ser mais importante do que outra, conforme a urgência dela. Um robô não pode continuar andando enquanto está caído, por exemplo. Por isso, o controle da câmera, posicionamento centralizado do robô com o objeto e o levantamento do robô são priorizados. Só quando estas condições são atingidas o robô poderá realizar outras funções como se aproximar do objeto. Esta forma de controle não é a mais rápida maneira de se aproximar de um objeto, mas é a maneira mais eficiente encontrada através dos testes realizados.

O sistema de controle hierárquico criado foi desenvolvido e baseado nas comunicações seriais dos dispositivos de forma sequencial. Com isso, o maior problema encontrado foi a criação desta rede de comunicação de forma confiável. O Arduino possui apenas um par Rx/Tx de comunicação serial assíncrona em hardware, o que não permite que haja comunicação com mais de um dispositivo ao mesmo tempo. Este problema havia sido resolvido através da inclusão de um circuito integrado multiplexador/demultiplexador 74HC4052 anteriormente, mas havia grande limitação com esta solução através de perda de informação. Com isso, houve a necessidade da criação de portas seriais desenvolvidas em Software. Isto foi possível através de uma biblioteca criada para o Arduino chamada de “Softwareserial”. Esta biblioteca funciona semelhantemente à comunicação serial por hardware, mas utiliza portas digitais em vez de entradas genuinamente seriais. Ela acrescenta interrupções externas de entrada, onde, quando um bit de início de mensagem é recebido, o microcontrolador entra em uma função de interrupção e guarda a informação recebida para posterior leitura.

O sistema de decisão de comando se baseia em regras de movimentação definidas. Quando o nível 2, ou nível de comando, decide uma ação ele envia um byte para MRC-3024 de forma serial, que representa a ação desejada. Esse código é representado pelo código ASCII de 0 a 8. Desta forma, cada código de ação criada pelo Arduino é realizado de forma semelhante a um ser humano quando utiliza um controle remoto para comandar o robô. O processo de decisão é feito baseado em lógica Fuzzy mostrado anteriormente e representa o processo de decisão semelhante a um ser humano. O controle hierárquico criado é feito de forma que as ações mais importantes, como o levantar após uma queda ou o ato de se posicionar corretamente antes de começar um movimento, são priorizados, ou seja, há hierarquia de decisões.

Os códigos são recebidos pela MRC-3024 e definem o movimento desejado. Esse movimento é realizado uma única vez e envia de volta um sinal, informando que o movimento acabou e que precisa de novas informações. Isso é necessário para que seja possível um controle e estabilidade de movimentos do robô e que o mesmo não execute movimentos indesejados. O conjunto de movimentos está presente na placa MRC-3024 de fábrica pela *Hitec*, o que facilita o trabalho de movimentação.

## **Resultados e Discussão**

Através de todas as modificações houve um grande avanço nas funcionalidades do Robonova-1, transformando-o em um robô inteiramente autônomo e capaz de realizar ações complexas para um robô humanoide. A introdução de uma câmera embarcada e o

desenvolvimento do novo sistema de controle, baseado na subdivisão de tarefas, trouxeram amplas possibilidades de controle e ações.

A melhoria do sistema de controle, assim como a criação de um algoritmo mais eficiente, levou o robô a ficar estável em suas decisões e movimentos. O trabalho procurou desenvolver um método de fácil entendimento e bem documentado para dar a possibilidade de prosseguimento neste projeto em anos futuros. Para isso, houve a criação do programa em ambiente do Arduino e um programa em Processing para haver uma integração entre o robô e uma estação de monitoramento em um computador. Outra possibilidade criada para facilitar o desenvolvimento de testes foi a possibilidade de controlar o robô remotamente ou deixá-lo autônomo através desta estação de monitoramento. Esta estação funciona de forma parecida com um supervisor, o que facilita o seu entendimento e utilização.

O uso dos sensores giroscópios e acelerômetros foram de grande utilidade na movimentação autônoma do robô humanoide. O sensor giroscópio utilizado funciona diretamente com a placa MRC-3024, onde internamente há um algoritmo de compensação através do giroscópio. Basta definir sua posição central e a placa já tem um programa para realizar a sua correção. O uso deste sensor melhorou a estabilidade do robô enquanto este se movimenta, mas, em determinadas ocasiões, o robô tende a ficar em uma posição fora do seu centro de gravidade quando parado. Isto pode ter ocorrido por falha no próprio sensor ou por algum problema interno de programação na placa MRC-3024. Este é um problema ainda não resolvido, mas que não gerou grandes problemas para o objetivo final, que era a movimentação autônoma do robô. O acelerômetro utilizado foi o responsável por definir a queda do robô nos momentos em que ela ocorre. Com ele foi possível saber para que lado o robô caiu e assim ter a possibilidade de levantá-lo.

Os sensores utilizados apresentaram alguns problemas indesejáveis na aplicação. O giroscópio apresentou alguns problemas de posição inicial depois de um período de testes e o acelerômetro se mostrou sensível a movimentação do robô, onde, mesmo quando não havia caído, ele marcava valores errados pelas acelerações impostas pelo robô ao se movimentar. Estes erros de medição levaram à queda do robô em algumas oportunidades, mas com a melhoria da programação e a realização de alguns filtros na programação, foi possível a correção de alguns destes erros. Uma melhor alternativa seria a introdução futura de um sensor que agregasse os dois sensores, como ocorre em uma IMU, explicada anteriormente. Com ela realiza-se a fusão sensorial necessária para o real posicionamento do robô de maneira mais eficiente.

A parte mais importante do trabalho foi a melhoria da comunicação serial entre os dispositivos e a criação de um canal de comunicação com o computador. O uso de portas seriais criadas em software foi de difícil configuração, mas, após diversas tentativas, houve sucesso na comunicação. O robô apresentou um tempo de resposta baixo para as decisões e obteve sucesso na maior parte dos comandos realizados. A aproximação do robô a um objeto ainda precisa de melhorias devido ao problema da câmera não dar valores muito confiáveis, o que não permite o robô se aproximar corretamente do objeto. A visão do robô, através da CMUCam4, evidenciou ser a parte mais frágil do projeto. Com a mesma houve problemas na detecção do objeto, o que é um problema na realização de testes. Muitas vezes a placa informava ter encontrado o objeto por ter objetos de cores parecidas no mesmo ambiente, o que deslocava a atenção do robô.



Futuramente haverá a necessidade de criar um sistema mais eficiente de processamento de imagens e que consiga rastrear mais de um objeto em uma mesma imagem. Paralelamente a este projeto, há outros projetos que visam esta mudança.

A seguir é mostrado o aspecto final do ROBONOVA-I após as modificações:

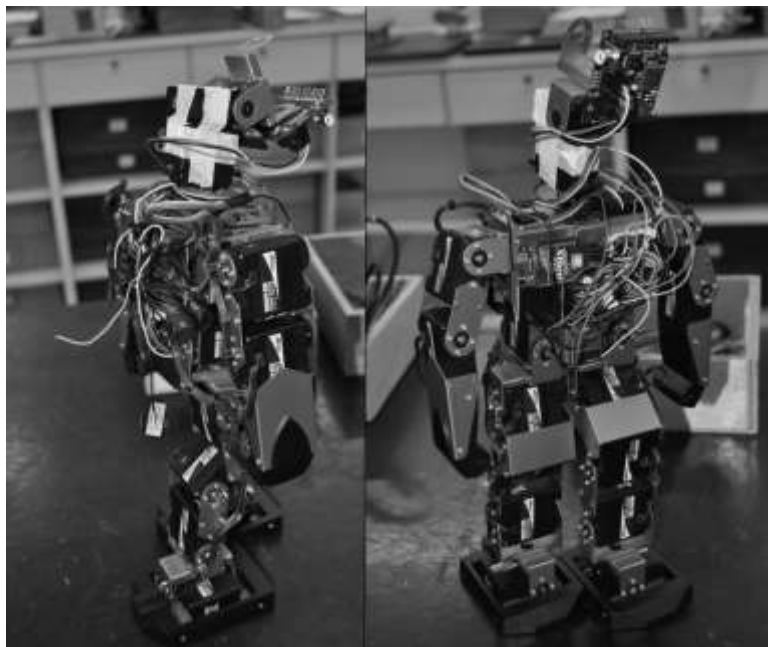


Figura 7 – Robô humanoide após modificações de hardware

## Conclusões

O projeto obteve resultados importantes para a conclusão dos objetivos propostos no início do trabalho. O robô é capaz de localizar um objeto e se movimentar através de seus sensores de forma autônoma. O trabalho ainda foi além e possibilitou a criação de uma estação de monitoramento externa para verificar o estado do robô e possibilitar um operador parar o robô quando desejar ou mesmo controlá-lo de forma remota.

A forma de controle baseada em uma estrutura hierárquica foi aperfeiçoada, permitindo assim maior precisão na tomada de decisão, assim como maior velocidade da mesma. A melhoria do algoritmo de decisão interno no Arduino possibilitou melhor aproveitamento da memória do mesmo, permitindo ainda a adição de melhorias de controle, podendo tornar a estratégia ainda mais complexa.

A pesquisa realizada neste trabalho foi baseada no aperfeiçoamento da estrutura de controle criada em trabalhos anteriores, na inserção dos sensores inerciais e criação de meios mais eficientes de comunicação serial entre os dispositivos internos e externos do robô. Os resultados demonstram que todos os objetivos neste quesito foram atendidos e o desenvolvimento de uma estratégia de controle mais eficiente é um trabalho que demanda tempo e experiência para alcançar e, com a continuação deste projeto no futuro, poderemos criar uma plataforma de pesquisa de robôs humanoides autônomos sólida.

O ponto mais importante criado neste trabalho foi o desenvolvimento de uma plataforma melhor de testes para haver um crescimento de complexidade nos trabalhos futuros. Todos os programas foram criados para um rápido entendimento para diminuir o tempo de procura de informações e houve sempre a preocupação por deixar todos os programas com o mínimo de erros possível. Toda a estrutura, sensores e programação estão com resultados satisfatórios e com grande possibilidade de aperfeiçoamento, o que é de grande utilidade para futuros trabalhos em robôs humanoides autônomos.

## **Referências Bibliográficas**

- Asada, M.; Ishiguro, H.; Kuniyoshi, Y.; MacDorman, K. F. (2001) Cognitive developmental robotics as a new paradigm for humanoid robots. *Robotics and Autonomous Systems*, V. 37, p. 185-193
- Assis, W. O.; Genova, W. J.; Gomes, M. M.; Coelho, A. D. (2007) Aplicação de Tecnologias de Controle de Processos Contínuos e Processamento de Imagens no Futebol de Robôs. *Anais do SBAI 2007 – Simpósio Brasileiro de Automação Inteligentes*, Florianópolis, SC, Brasil.
- Crétual, A.; Chaumette, F. (2000). Dynamic stabilization of a pan and tilt camera for submarine image visualization. *Computer Vision and Image Understanding*, V. 79, n. 1, p. 47-65
- Cui, X. T.; Moran, D. W.; Schwartz, A. B.; Weber, D. J. (2006) Brain-Controlled Interfaces: Movement Restoration with Neural Prosthetics. *Neuron*, V. 52, p. 205-220.
- Gonzales, R. C., Woods, R. E. (2000) *Processamento de Imagens Digitais*. Edgard Blücher Ltda.
- Kikuchi, D. Y. (2007). Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência. Tese (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, SP.
- Nicholls, E. (1998). Bipedal Dynamic Walking in Robotics. Honours Thesis – The University of Western Australia; Department of Electrical and Electronic Engineering.
- Palep, J. H. (2009). Robotic Assisted Minimally Invasive Surgery. *Journal of Minimal Access Surgery*. V.5, I.1, p. 1-7.
- Rowe, A. (2003) ROBONOVA-I – 16 Servo Entertainment Robot Box Kit – Introduction Manual. Edited by Charles Rosenberg and IllahNourbakhsh, Carnegie Mellon University.