# Portland State University

# Department of Electrical Engineering

# Portland Cyber Theatre Group

A pre-proposal to Dean of MCECS for a Guide Robot



Marek Perkowski, Mathias Sunardi, Jim Larson, Robert Fiszer, Michael Lowe

[mperkows@ece.pdx.edu](mailto:mperkows@ece.pdx.edu), msunardi@ece.pdx.edu

December 15, 2011

# Executive Summary

This is the pre-proposal to Dean of MCECS at PSU to develop an intelligent autonomous robot Guide that will give guided tours and be able to lead a user to a sequence of specific locations in the basement area of the Engineering Building and the Fourth Avenue Building at Portland State University. For the purpose of this proposal we will call this robot MCECS-BOT (MCGuide, McGuideBot, or else, the name to be changed for one proposed by the Dean). The robot will look like a healthy young person controlling a scooter. The robot will be in a standing position and will have a head with face and a pair of arms to express emotions. Guests will be able to interact with MCECS-BOT using speech recognition and speech synthesis, vision-guided gestures and facial expressions of the robot, and a touchscreen. They will ask to be guided or directed to a particular classroom, location, laboratory or office within the basement floor of the EB and FAB. Once MCECS-BOT understands the person’s request, it will autonomously and safely navigate the building to reach the requested location. Of additional importance, MCECS-BOT will be able to communicate in a subset of English and will know also standard greetings in Chinese, Spanish, German and other languages of potential visitors. It will advertise our great engineering programs and answer standard questions like sizes of classes, or types of degrees awarded. The robot will be able to detect the presence of a person using the face detection and tracking algorithms, and also navigate the hallways safely by avoiding people and other obstacles using sonar, infra-red sensors, and input from two Kinect vision systems. The revolutionary recent product from Microsoft – the Kinect peripheral, will be used as the main vision input. At first, knowledge of the basement area of the Engineering Building/Fourth Avenue Building will be manually programmed. By the end of the project, an automatic mapping algorithm will be implemented such that MCECS-BOT would be able to build its own map and explore new areas. Extension of the robot’s software is also possible such that the robot will be also to use the elevators, open doors, and visit other floors. MCECS-BOT will engage its user in a conversation using spoken English, albeit a simplified one. This conversation will be also related to self-diagnostics of the robot hardware/software system. Conversely, users can interact with and give commands to MCECS-BOT using speech by means of a set of keywords and simple English sentences. For the head of the robot we propose a mobile tablet (in addition to the touchscreen) that will display an illustrated/cartoon character face showing multiple facial expressions or a puppet/Muppet-like face with one or more degrees of freedom for facial expressions. The head will be attached to the torso by a neck with three degrees of freedom. Together with the movements of the neck/head and torso, the facial expression will make the interaction with MCECS-BOT more engaging to the user.

The MCECS-BOT project will be executed under the supervision of Dr. Marek Perkowski from the Electrical and Computer Engineering Department as part of the Portland Cyber Theatre project. The project is estimated to take 18 months of hardware and software development to completion. It will be developed in phases, each phase of 6 months, and the results of each phase will be demonstrated to the Dean and his staff who will be asked eval questions and asked for additional feedback. The core project team will consist of Ph.D., MS, undergraduate and high school students that are interested in the areas of artificial intelligence, computer vision, and autonomous robotics. The project will be completely connected to the PAVE frame on Internet (developed by Prof. Fei Xie from CS and his team). This way, our robot and PSU labs will be visible from the entire world and viewers from other countries will be able to control the robot. In the last 6 months we have done some preliminary studies for this project and developed software, so we are now sure that the presented here project proposal is realistic.

For the realization of the MCECS-BOT project, we expect the total costs to be $17,214. Which includes hardware, software, and stipend for student developers and the student manager (Mathias Sunardi, a Ph.D student of Dr. Perkowski). Among the hardware needed are a mobile platform (a scooter, around $2000), a mainboard computer for the robot’s main controller, Microsoft’s Kinect (already purchased and must be soon reimbursed to the student), two tablets, sensors for navigation and interaction, many servo motors, and materials for the robot human-like body (including the tablet-head or realistic face with an expensive latex mask if we want the face to realistic – depends on the selected variant). For software, we will utilize open-source software as much as possible. MCECS-BOT will run under Ubuntu Linux Operating System. The vision system will be controlled using the OpenCV image processing library, the speech recognition system will use the Pocket-Sphinx software from Carnegie Mellon University, the speech synthesizer system will use FreeTTS software, and the control of all those components will be synchronized using a system which will be developed by the project team. The integrating system will consist of the new robot architecture, the learning system, the navigation and mapping system, the adaptive behavior, and novel approaches for generating new expressive behaviors and facial gestures. The design of this robot will have not only engineering aspects that are required for a practical robot to be fully operational in MCECS, but also the purely research components such as the use of higher order logic, motion generation and pattern recognition aspects related to the main research areas of Dr. Perkowski’s Intelligent Robotics Lab.

## Vision

To showcase the awesomeness and ingenuity of ongoing robotics research and class projects that has been done by PSU MCECS students, while promoting Portland State University as a formidable research center in intelligent robotics technologies among leading universities in the field such as MIT, Carnegie Mellon University, and Stanford. So far, only top U.S. universities have built this kind of robots, and our robot will be more advanced in the area of expressing its “emotions”, language communication and recognizing emotions of the visitors.

## Mission

To have a fully functional, stable, and reliable autonomous robot platform that acts as a guide for guests to the MCECS. Additionally, MCECS-BOT would be the primary platform for human-robot interaction research, particularly in the application of: Robot Theatre, tele-presence, assistive robotics, and social robotics. Such applications involve researches in: computer vision, autonomous navigation, natural language processing, speech recognition and synthesis, machine learning, kinematics, and aesthetics. Moreover, because of the breadth of the robotics research, MCECS-BOT could also serve as a platform for multi-disciplinary research collaboration projects.

# Table of Contents

Executive Summary ------------------------------------------------------------------------------------------------------------- 2

Table of Contents ----------------------------------------------------------------------------------------------------------------- 4

Portland Cyber Theatre -------------------------------------------------------------------------------------------------------- 5

Prototype – The GuideBot Project ------------------------------------------------------------------------------------------- 6

The MCECS-Bot ------------------------------------------------------------------------------------------------------------------- 8

* Vision ----------------------------------------------------------------------------------------------------------------------- 10
  + Microsoft Kinect ---------------------------------------------------------------------------------------------- 10
* Navigation ---------------------------------------------------------------------------------------------------------------- 11
  + Mobile Platform ---------------------------------------------------------------------------------------------- 11
  + Sensors ---------------------------------------------------------------------------------------------------------- 12
* Speech ---------------------------------------------------------------------------------------------------------------------- 12
  + Microphone Array -------------------------------------------------------------------------------------------- 12
  + Pocket-Sphinx ------------------------------------------------------------------------------------------------- 12
  + FreeTTS --------------------------------------------------------------------------------------------------------- 13
* Core Processing Unit --------------------------------------------------------------------------------------------------- 13
  + Vision Processing --------------------------------------------------------------------------------------------- 13
  + Natural Language Processing ---------------------------------------------------------------------------- 14
  + Navigation ------------------------------------------------------------------------------------------------------ 14
  + Behavior --------------------------------------------------------------------------------------------------------- 14
* Integration ----------------------------------------------------------------------------------------------------------------- 15
* Computer ------------------------------------------------------------------------------------------------------------------- 15
  + Motherboard ---------------------------------------------------------------------------------------------------- 15
  + Memory ----------------------------------------------------------------------------------------------------------- 16
  + Hard Drive ------------------------------------------------------------------------------------------------------- 16
  + Wireless Extender Antenna -------------------------------------------------------------------------------- 16
  + Power Supplies/Batteries ----------------------------------------------------------------------------------- 16
* Robot Construction ------------------------------------------------------------------------------------------------------ 17
  + Modification to the Mobile Base -------------------------------------------------------------------------- 17
  + Building the Robot Body ------------------------------------------------------------------------------------ 17
  + Building The Robot Arms ----------------------------------------------------------------------------------- 17
  + Building The Head -------------------------------------------------------------------------------------------- 18

Estimated Cost ----------------------------------------------------------------------------------------------------------------------- 19

Project Timeline Estimate -------------------------------------------------------------------------------------------------------- 19

References ----------------------------------------------------------------------------------------------------------------------------- 23

Appendix A ---------------------------------------------------------------------------------------------------------------------------- 24

# Portland Cyber Theatre

The Portland Cyber Theatre is a brainchild of Dr. Perkowski who envisioned a theatrical play where all or the main characters are played by Robot Actors. The Theatre is a good example of a complex, multi-faceted robot application. Particularly, the three main research areas involved are: computer vision (face detection, object detection), navigation (shortest path, mapping, obstacle avoidance), and human-robot interaction (speech recognition, natural language processing, speech synthesis, symbolic gestures, animation). Several of Dr. Perkowski’s students have developed and built robots towards this goal in the past. The debate play between Einstein and Schrödinger’s Cat robots built by Dr. Perkowski’s high school students, Arushi Raghuvanshi and Sidharth Dhawan (Figure 1) [6],a motion and event editor based on Brzozowski’s derivation method [1], facial recognition using Principal Component Analysis and neural networks [2], and generation of rhythmic and expressive robot motion from MIDI music [3]. Also, novel approaches employing well-known methods in logic synthesis [8, 13], and quantum computing [7,8,11] have been explored to develop robot behaviors in the context of Robot Theater.

(a) (b)

Figure 1 a) Einstein robot by Sidharth Dhawan, b) Schrödinger’s cat by Arushi Raghuvanshi

Several leading robotics research teams and even some commercial companies have also tinkered with similar ideas in the past. Dr. Cynthia Breazeal and the Media Lab from MIT have created an interactive theatre of an anemone-like robot which the audience can affect its behavior by their proximity to the robot [4]. Norman Badler from University of Pennsylvania has done many works on incorporating the movement theory Laban Movement Analysis (LMA) into control systems for 3D models [5]. Bluebotics SA from Switzerland worked together with Christian Denisart, a Swiss Theatre director to create a play called Robots[[1]](#footnote-1) (Figure 2a). The play involves humans and robot actors. Osaka University in Japan also created a short play called Hataraku Watashi[[2]](#footnote-2) using the Wakamaru robot from Mitsubishi, in addition to several human actors (Figure 2b).

(a) (b)

Figure 2. a) A scene from the play Robots, b) a scene from the play Hataraku Watashi.

Despite the achievements of the previous PSU student projects of showcasing individual or very few elements of the Robot Actor, individually, these projects are not able to show the complete experience of a Robot Actor. These elements need to be integrated into one system and exhibited by a robot. With MCECS-BOT, the Robot Actor can be finally embodied in one robot.

## Prototype – The GuideBot Project

Currently, a prototype of MCECS-BOT is being developed as a class project called ‘GuideBot’ by a group of students from the ECE 478/578 Intelligent Robotics class taught by Dr. Marek Perkowski. Five students are involved in this project: Jim Larson, Jules Alfani, Robert Fiszer, Mike Lowe, and Hamed Mirlohi. Our current Guide-Bot robot is shown in Figure 3, and the project itself consists of several components:

* Microsoft Kinect sensor for vision
* Pioneer Robot base for mobility
* Pocket-Sphinx speech recognition software
* OpenCV image processing software
* Mobile Robot Programming Toolkit (MRPT)
* FreeTTS text-to-speech software
* Zotac MiniATA motherboard with Linux operating system installed in a hard drive on the robot.

A group of students from the ECE478/578 class in Fall 2010 have previously built a similar guide robot project called ‘PeopleBot’. The PeopleBot project was able to achieve several key features:

* Using Pocket-Sphinx, it was able to recognize a set of commands from speech from the user
* It was able to carry out simple conversations with the user, powered by the A.L.I.C.E. (Program D) chatbot program.
* It was able to verbally respond to the user by converting the text output from A.L.I.C.E. to speech using the FreeTTS program.
* Fuzzy Logic was implemented to control the behavior of the robot based on input from face detection, voice input, and the internal states of the robot.

The GuideBot project aimed to improve upon the PeopleBot project by:

* using Microsoft Kinect to obtain more information from vision to enable more complex behavior for the robot, and help with robot alignment during navigation,
* using better natural language processing (NLP) to allow the robot to distinguish and understand commands, questions, and statements,
* giving the robot more ‘character’ in its behavior and conversations,
* and reliable navigation system so the robot can navigate the Engineering Building in a reliable and safe manner, by avoiding obstacles and keeping it from running into the hallway walls.

At its current state, the robot is able to navigate between pre-defined checkpoints (Intelligent Robotics Lab, Men’s room, Ladies’ room, Tektronix Lab, PAVE Lab). The robot also demonstrated an obstacle-avoidance behavior and able to return to its original path as seen in Figure 4. The GuideBot team was able to interface the Kinect under Ubuntu Linux, and an NLP module written in Java was demonstrated to be able to distinguish a sentence (text input) as a statement or command. Speech recognition has not been achieved.



Keyboard + trackpad

Pioneer 2:

* mainboard
* battery
* sonar array (16)
* 2 DC motor
* bumpers
* shaft encoders

Motherboard + HDD:

* Linux 10.10
* MRPT
* OpenCV
* Pocket-Sphinx
* NLP
* FreeTTS

Monitor

Kinect

Figure 3. The GuideBot robot, first prototype, December 2011.

## 

Figure 4. The GuideBot in action – avoiding an obstacle (the bag) and returning to the original path.

MCECS-BOT will physically be a separate robot from GuideBot. However, the software for the key features that have been developed on GuideBot were designed such that it can be directly applied on MCECS-BOT with minimal reprogramming (with the exception of motion control of the base). At its current state, the GuideBot project includes only some subset of components of the complete MCECS-BOT and will be demonstrated in Winter quarter 2012. GuideBot project will be continued in ECE 479 class in Winter 2012 and the final demo will be in March 2012.

# The MCECS-BOT

Figure 5 shows the overview of MCECS-BOT system. A typical scenario of interaction from a hypothetical user named Johnny with the MCECS-Bot is as follows. Johnny walks into MCECS-Bot’s view and say “Hello, MCECS-Bot!” The microphone array in the first Kinect receives his voice and sends it to the Pocket-Sphinx to be quantized as a sequence of keywords. At the same time, the Kinect cameras capture the sequence of images of Johnny, and send them to the Vision unit (OpenCV/OpenKinect/MRPT block) to locate a face, try to identify if the face belongs to a human, a known person, and if it can detect any identifiable gesture from the person. Let’s assume that the system is able to identify the face as belonging to a person named Johnny, and no gesture is detected. The name ‘Johnny’ is then passed to the NLP. The resulting sequence of keywords from Pocket-Sphinx is passed to the NLP unit, which then determines that this sequence is a statement; specifically, a greeting statement. The NLP then retrieves the appropriate greeting response “Hello, -“ and combines it with the name “Johnny” it received from the Vision unit, to create the sentence “Hello, Johnny!” The sentence is then passed to the FreeTTS to be synthesized into acoustic voice. Concurrently, as the system generates the “Hello, Johnny!” sentence, the Behavior unit also tries to retrieve an appropriate gesture response. NLP tells the Behavior unit that a greeting was identified, so a waving hand (greeting) gesture should be executed. The Vision unit tells where the person is, which tells the Behavior unit to orient the robot to face the speaker. Thus, the robot will orient itself to the speaker, wave an arm, and say “Hello, Johnny!”

The MCECS-Bot is now in the mode to accept more input from the user Johnny. Johnny can then interact further with MCECS-Bot by speaking what he wants from the robot. For example, “Go to the Tektronix Lab.” The spoken sentence will once again processed by Pocket-Sphinx and the sequence of keywords is analyzed by NLP. The NLP identifies this sentence as a command to navigate to the Tektronix Lab and sends this information to the Behavior unit. Meanwhile, the robot will first confirm with Johnny whether or not it understood the command correctly. Once confirmed, the Behavior unit will then issue a command to the Navigation unit, the robot will ask Johnny to follow it, and then starts navigating to the Tektronix Lab. A second Kinect is used to help the robot navigate by detecting walls and other obstacles.

Visual feedback is given through the Tablet/Touchscreen. One Tablet/Touchscreen will be used to show the face of the robot with various facial expressions (like in [15]), and system diagnostics. The second Tablet/touchscreen is used to display a user interface for the user to provide input such as commands, questions, manual controls, and help on how to interact with the robot. Speech recognition is often hit-and-miss, so a medium for the user to give inputs manually to the robot is provided using the Tablet/Touchscreen.

Kinect #1

Hello,

MCECS-Bot!

Speaker

Hello,

Johnny!

Kinect #2

Tablet/ touchscreen



Johnny,

Guest

Tablet/ touchscreen



Robot body, head, arm, base

MCECS-Bot Core Processing

OpenCV

OpenKinect

MRPT

Pocket-Sphinx

NLP

FreeTTS

Navigation

Behavior, head/arm/ body gestures

Diagnostics, User Interface, facial expr.

hallway

Face,

Gestures

Figure 5. MCECS-Bot system.

Below we describe the various components of MCECS-Bot and how we intend to use them.

## Vision

### Microsoft Kinect ($149.99)

The Kinect is a vision-based peripheral by Microsoft for the XBOX 360 video game console, to be used for the players to control video games using his/her gestures and/or movements (Figure 6). Kinect uses a combination of a RGB camera, a depth sensor, and microphone array as inputs that are translated as controls to the XBOX 360. The RGB camera is used to capture color images and face detection. The depth sensor is a combination of infra red array projector and a CMOS monochrome camera that could detect the position and distance of a surface (an object or the person’s hand and body) from the sensor. The microphone array can be used for acoustic source localization. Not long after its launch, hobbyists were able to use the Kinect for many other applications beyond just video game on the XBOX 360. Recognizing the potential and the enthusiasm of the community with the Kinect, Microsoft recently also released a Kinect Software Development Kit (SDK) to allow the community to innovate using Kinect. (Our goal is to use Kinect in all our current and future robotics projects instead of standard web cameras and microphones used so far).

Currently, a community-developed, open-source Kinect SDK aptly called OpenKinect is available for the Windows, Linux, and Mac platforms[[3]](#footnote-3). In addition, a collection of C++ libraries called The Mobile Robotic Programming Toolkit (MRPT) also has support for Kinect. MRPT is also an open-source (free) library. We are still evaluating between using the OpenKinect SDK and MRPT as the best tool to develop the vision system for MCECS-BOT.

The MCECS-Bot will be using two Kinects; one facing the forward/path to travel and one facing the guests following it. The one facing the path will be used to aid the robot in navigating around the hallways, while the one facing the guest will be used to get input about the guest such as gestures, faces, and presence of the guest. Some additional fixtures with one degree of freedom will be used to control the Kinect that is being used to track the guests.



Figure 6. Microsoft Kinect

## Navigation

### Mobile Platform (approx. $2000)

The robot will be roaming on a stable platform with motorized wheels. We anticipate a lot of time to be spent on modifying the mobile base with sensors and designing the interface to control it with our system. Shown here are some examples of the types of mobile base we envisioned. Figure 7 is a model of a simple electric scooter, while Figure 8 shows the Segway RMP100, a popular model used in many robotics project such as was done by Intel and Carnegie Mellon University but its cost is about eight times of the former.



Figure 7. A candidate for our mobile base – a personal electric scooter[[4]](#footnote-4).



Figure 8. Segway RMP 100 (Robotic Mobility Platform)[[5]](#footnote-5)

### Sensors (various)

We need to add sensors to the mobile base for safe and reliable navigation. We plan to use sonar array like in the Pioneer 2 base: 8 sonar sensors in front and 8 sonar sensors in the back, each side arranged in a half-circle array. Additionally, we would like to add infrared sensor arrays to attach to the bottom of the base to detect edges, or lines on the ground that may be markers of the boundaries of a room. We also plan to install touch sensors and infrared sensors to the robot arms for safety measures. For example, the robot arm will retreat or stop moving if it hits something unexpectedly.

## Speech

### Microphone Array (Included in the Kinect)

We will use the Kinect microphone array for acoustic source localization, and as speech input from the users. The speech input will be processed by the Pocket-Sphinx (or Sphinx4) speech recognition software to distinguish keywords in the speech. The input from the microphone array will also be used to detect if a person is nearby the robot, and to determine the person’s location relative to the robot.

### Pocket-Sphinx (Free, Open-source)

Pocket-Sphinx is a version of the Sphinx speech recognition software that can be used in embedded systems (e.g., based on an ARM processor). Pocket-Sphinx is under active development and incorporates features such as fixed-point arithmetic and efficient algorithms for GMM (General mixture model) computation. In statistics, a mixture model is a probabilistic model for representing the presence of sub-populations within an overall population, without requiring that an observed data-set should identify the sub-population to which an individual observation belongs.

The Pocket-Sphinx will be using word-spotting to identify certain keywords in the user/guest’s speech that is known to the MCECS-Bot (i.e. in its vocabulary. See the section on Natural Language Processing). The sequence of keywords will then be passed to the NLP unit to be analyzed.

An alternative to Pocket-Sphinx, a commercial software called Dragon Naturally Speaking by Nuance is also available. However, there has not been enough evidence on the advantage of the Dragon software over open-source software such as Pocket-Sphinx in terms of cost over features, speech recognition performance, and ease of interfacing with the software. Moreover, the Dragon software only works under Windows OS and does not offer support for 64-bit operating systems. Thus, at the moment it is not of our interest.

### FreeTTS (Free, Open-source)

FreeTTS is an open source text-to-speech synthesizer, written in JAVA™ programming language. FreeTTS is based upon Flite, a small run-time speech synthesis engine developed at Carnegie Mellon University and is built by the Speech Integration Group of Sun Microsystems Laboratories. It uses Hidden Markov Model (HMM) for synthesis. It provides partial support for the Java Speech API (JSAPI).

The FreeTTS will be used to provide speech feedback to the users/guests. FreeTTS takes text as input, and ‘reads’ the text as speech. The text responses will be the results of the Natural Language Processing unit from the speech input the user/guest provided.

## Core Processing Unit

The Core Processing unit consists of four components: a) Vision processing, b) Natural language processing, c) Navigation processing, and d) Behavior processing. These components will be realized using the MRPT, OpenKinect, and OpenCV libraries, in addition to the software that we will develop to integrate them.

## Core Processing Unit – Vision Processing

As mentioned above, the MCECS-Bot will use Microsoft Kinect as its primary vision/image sensor. The vision input will be used for the following:

* Facial recognition. The MCECS-Bot will learn the faces of the Dean and faculty members and staff in MCECS so it can identify and greet them accordingly. Additionally, if the MCECS-Bot is in the company of guests, it can introduce the Dean, faculty members and/or staff to the guest if they are present.
* Navigational aid. In addition to the sonar, the input from vision will provide additional information for the robot to navigate the hallways safely. Mainly, vision will provide information on the distance between the robot and an object, person, or obstacle in front of it, and the relative position and alignment of the robot’s path with the hallways. Also, using vision the robot may be able to identify ‘landmarks’ near it and determine its current location. Using this information, the robot can better avoid obstacles, and remain on its path without running into the hallway walls.
* Gesture recognition. The Kinect can be used to identify hand gestures of visitors. A set of simple gestures will be available to users to control the MCECS-Bot such as stopping, greeting, or direction.

The interface to the Kinect peripheral will be done using the OpenKinect C/C++ library or the MRPT library. Both are free, open-source libraries.

## Core Processing Unit – Natural Language Processing

The Natural Language Processing (NLP) unit is able to distinguish sentences as commands (e.g. “go to the robotics lab”), questions (e.g. “where am I?”), or statements (e.g. “I am sad”). The NLP unit was developed by Robert Fiszer for his part in the GuideBot project. The sentences are obtained from the speech recognition system (Pocket-Sphinx) using word-spotting, in the form of a sequence of keywords. A sentence and its sequence of keywords are analyzed using a subset of the English grammar. If a sentence can be categorized into one of the three types above using the grammar, then the sentence is valid. Once the NLP unit determined that a sentence is valid, a corresponding response is returned to the user in the form of an action (if the sentence was a command), an answer (if the sentence was a question), or a statement (if the sentence was a statement).

The NLP unit will be improved by:

* Adding more complex grammar,
* Adding more words in the vocabulary,
* Adding responses for the user such as greetings, inquiries, and explanation about the MCECS and about the MCECS-Bot itself.

## Core Processing Unit – Navigation

The Navigation unit controls how the robot navigates the area. Specifically, it maintains the map of the area, controls the motors driving the wheels of the robot to avoid obstacles, making turns, and maintain a safe distance to the hallway walls using input from sonar and vision.

The navigation unit of the MCECS-Bot will improve upon the navigation program on the GuideBot robot written by Michael Lowe. From the GuideBot demonstration, we identified the following to be improved:

* Use the sonar arrays to make the robot keep a certain distance from the hallway walls.
* Use shaft encoders to read the distance traveled by the robot.
* Use shaft encoders to read the distance traveled by each wheel such that the information can be used to detect when the robot is not traveling in a straight line, and make necessary adjustments,
* Make the robot turns in an arc from a straight path.

The navigation unit will be written by the project team and utilizing some functions from the MRPT library.

## Core Processing Unit – Behavior

The Behavior unit is essentially the ‘brain’ of the system. Inputs from the NLP and Vision unit are processed here to understand the current situation external to the robot, and determine how the robot needs to respond to the situation. By ‘situation’ we mean information about the user (e.g. position, presence, commands) and the current environment of the robot. The Behavior unit will make the robot perform gestures to accompany the speech responses. The robot will perform simple gestures such as hand waving, pointing, turning around, and a little dance sequence. It is here where we will implement the intelligence for the robot, or simply the mapping of the input from the user and environment to the speech, display, and action outputs of the robot.

We are planning to implement extensions of the works done in past student projects, such as [1,2,3] to enhance the user experience with the MCECS-Bot. At some point, a program will be implemented to enable the robot to synthesize its own behavior based on inputs from the user and the environment, among other things.

## Integration

All of the above components – vision, speech, navigation, input from sensors – would be integrated using a program that will be developed by the project team to create the system described in Figure 5. From our experience from past projects, the work to integrate the components usually takes the majority of the software development time. Often, each team member prefers one programming language while other members may be comfortable programming in a different language. Moreover, often installation of software libraries required for the project runs into roadblocks due to many factors, such as unmet dependencies, unsupported versions, deprecated functions, and other issues that was not addressed in their documentation. Thus, we anticipate much of the time allocated for software development will be spent on the integration, testing, evaluation and incremental improvement work.

To help ease the integration work, we decided to use the C/C++ programming language to develop the low-level functionalities of the robot (e.g. navigation, speech recognition, vision, servo/motor controls), and use the Lisp programming language to program the higher-level functionalities in the Behavior unit. We believe the C/C++ language is appropriate for the low-level functions of the robot since many of the available libraries to be used was written in C/C++. Since Lisp is well suited (and still preferred today) for programming algorithms used to solve complex problems such as search, and artificial intelligence in general, we want to use it to implement ‘intelligence’ in the MCECS-Bot.

## Computer

### Motherboard ($215)

We will use the same model as was used for the GuideBot project; the Zotac IONITX-L model (Figure 9). A specification of the motherboard is attached in Appendix A.



Figure 9. Zotac motherboard

### Memory ($20)

4GB DDR3 RAM (DIMM) for the motherboard.

### Hard Drive (approx. $100)

We will need a hard drive to install the Ubuntu Linux OS and all the programs we will use.

### Wireless Extender Antenna ($40)

Like in the GuideBot project, we will add a wireless range extender antenna to the motherboard for the robot to have better wireless connection.

### Power Supplies/Batteries (approx. $156 ea.)

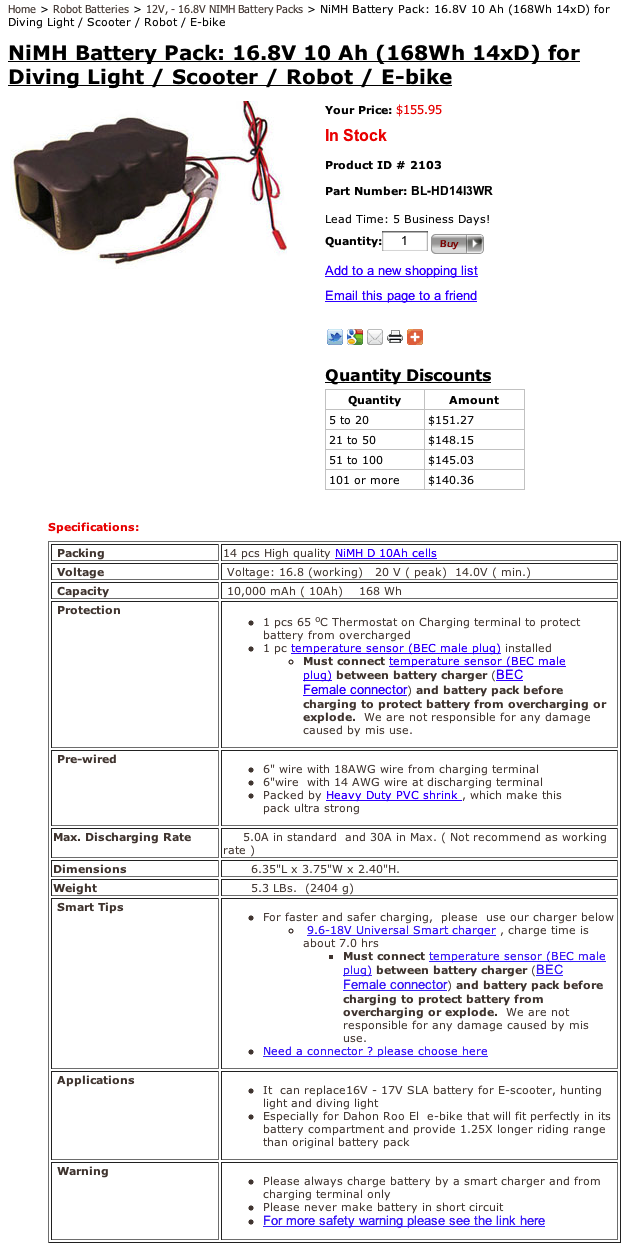


Figure 10. Battery pack to power computer and other servos besides the mobile base.

# Robot Construction

The construction work of the robot consists of four parts:

* Modification to the mobile base
* Building the robot body (torso/upper body and lower body)
* Building the robot arms
* Building the head

## Modification to the Mobile Base

The mobile base needs to be modified to create placement mounts for the robot body. The robot body will be fixed to the mobile base using brackets and a rigid aluminum pole to support the body. In addition to the mounts for the robot body, mounts for power supply (battery packs) and sensors will also be needed on the mobile base. Hopefully we will not need custom-made brackets.

## Building the Robot Body

The body of the robot consists of two parts: the upper body or torso, and the lower body. The torso will consist of mounts for the neck and two arms, and will be connected to the lower body with two degrees of freedom on the ‘waist’. The degree of freedoms on the ‘waist’ is in place to allow the robot to a) show gestures, and b) dance. The top part of the torso will be attached to the pole erected from the base so the body will be fixed to the mobile base. The upper part of the lower body will be able to freely move with respect to the torso, but the bottom part remains attached to the mobile base. We will use aluminum frames to build the ‘skeleton’ of the body. The torso will also have fixtures to place the touchscreen, motherboard, hard drive, Kinect, and range sensors (sonar). For presentation, the body may be covered using cloth, or plastic shells. The touchscreen and Kinect must remain exposed, however.

## Building the Robot Arms

Each of the robot arms will have two degrees of freedom at the shoulder, one degree of freedom at the elbow, and two degrees of freedom at the wrist (Figure 11). For now, the hand will be in a fixed, relaxed, open-palmed pose (instead of a clenched fist). The home position of the arms will be holding the handlebars of the mobile base. Occasionally, the arms will be used to perform gestures such as waving arm, pointing to a direction, ‘follow me’, or dancing. Similar to the body, aluminum frames will be used to create the structure of the arm, and for presentation, the arm may be covered using cloth or plastic shells. Touch and proximity sensors will be added to the arms for safety measures.

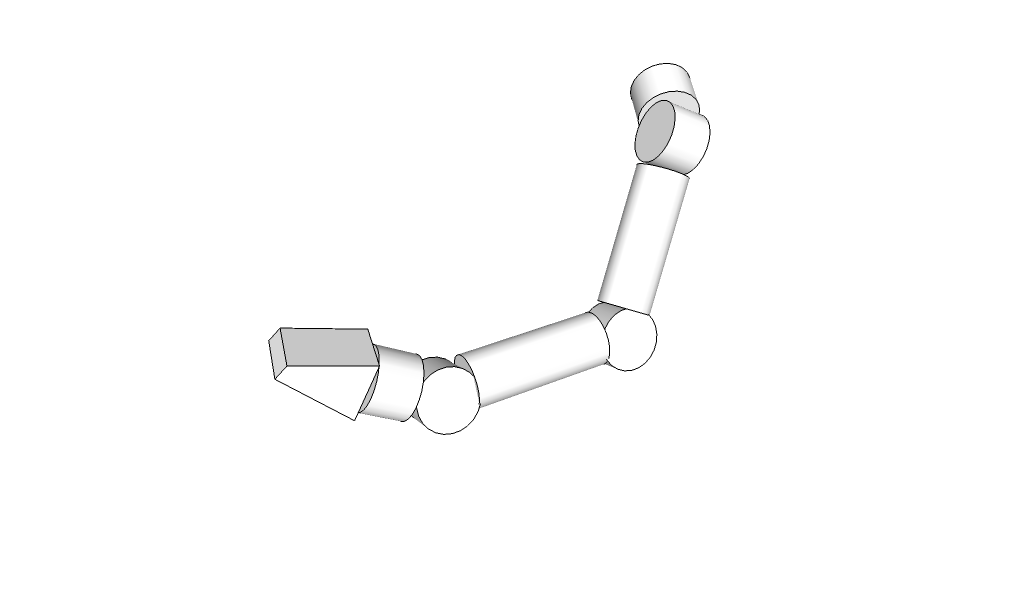


Figure 11. Arm and hand degrees of freedom. Bottom left is the hand (prism-shaped), and top right is the shoulder.

## Building the Head

The head of the robot will be a mount for the second tablet/touchscreen, connected to the torso with a neck with three degrees of freedom (yaw, pitch, roll for the head).

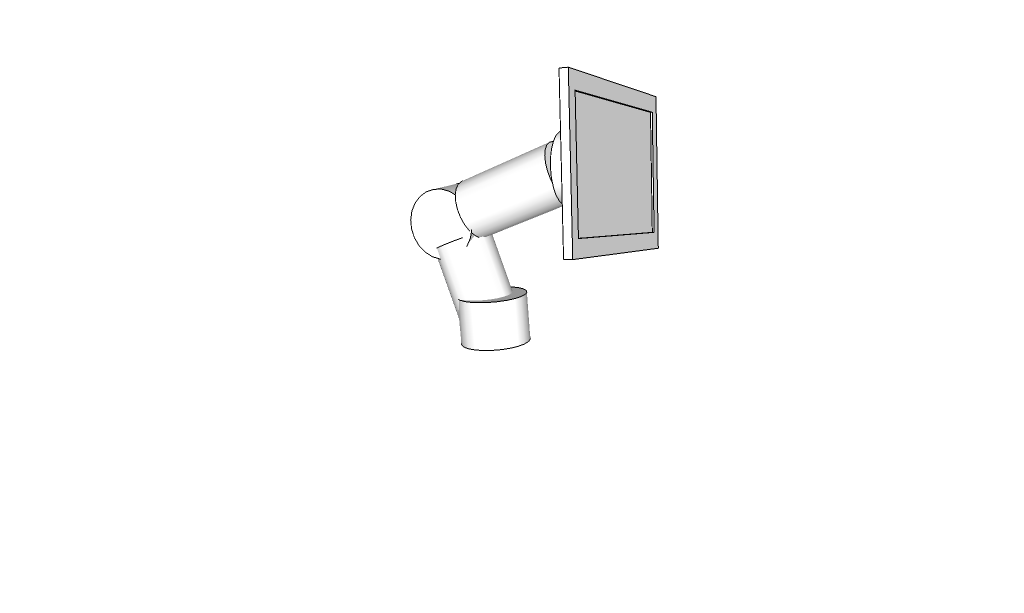


Figure 12. Degrees of freedom on the neck/head. Base of the neck (yaw), middle of neck (pitch), and top/face (roll).

# Estimated Costs

The following are our projections on the cost to build the MCECS-Bot

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Retail Price (US$) | Quantity | Cost |
| Mobile Platform / Scooter | 2000.00 | 1 | 2000.00 |
| Motherboard | 215.00 | 2 | 430.00 |
| RAM | 20.00 | 1 | 20.00 |
| Hard Drive | 100.00 | 2 | 100.00 |
| Wireless extender antenna | 40.00 | 1 | 40.00 |
| Batteries | 156.00 | 4 | 624.00 |
| Microsoft Kinect | 150.00 | 2 | 300.00 |
| Servo | 50.00 | 10 | 500.00 |
| Tablet | 350.00 | 2 | 700.00 |
| Sensors | Varies | Varies | 700.00 |
| Building material (sheet metal, screws, etc.) | Varies | Varies | 1000.00 |
| Stipend | 25.00/hr | 432 hours (for 18 months) | 10800.00 |
| Total |  |  | 17214.00 |

# **Project Timeline Estimate**

The following is our estimate of the amount of work items involved.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | Project | Description | Complexity | Notes |
| Winter 2012 | Vision 1 | Use Kinect to track faces using face detection | High |  |
|  |  | Add face recognition to face detection | High |  |
|  |  | Build library for integration with the system | Medium |  |
|  | Vision 2/Navigation | Use Kinect to detect hallway walls/edges | High |  |
|  |  | Use information of hallway walls/edges to orient robot | Medium |  |
|  |  | Build library for integration with the system | Medium |  |
|  | Speech | Train Pocket-Sphinx for 85%+ detection rate | High |  |
|  |  | Interface Pocket-Sphinx with NLP unit | Medium |  |
|  |  | Build library for integration with the system | Medium |  |
|  | NLP | Improve current NLP program | High |  |
|  |  | Improve grammar database | Medium |  |
|  |  | Add new words to the vocabulary | Low |  |
|  |  | Add responses to user commands | Low |  |
|  |  | Add responses to user questions | Low |  |
|  |  | Add responses to user statements | Low |  |
|  |  | Interface with FreeTTS and behavior unit | Medium |  |
|  |  | Build library for integration with the system | Medium |  |
|  | Body Construction | Obtain and modify mobile base | Medium | Investigate how to interface with the base, design and make mounts for sensors and robot body |
|  |  | Start design on robot body | Medium | Torso and lower body mechanisms |
| \*\*\*\*\*\*\*\*\*\* | Winter 2012 Deliverables | Library for face detection and recognition, library for hallway walls/edge detection, library for speech recognition, library for NLP, mobile base is prepared for mounting sensors and body, body design | \*\*\*\*\*\*\*\*\*\* |  |
| Spring 2012 | Vision | Recognition of user’s arm and body gestures | High | Using Kinect |
|  | Navigation | Wall following | Medium | Using information from Kinect and sonar arrays |
|  |  | Smooth turning | Medium |  |
|  |  | Improve obstacle avoidance | Medium |  |
|  | Behavior | Program some simple gestures for conversation | Low |  |
|  |  | Map some input combinations to some behaviors | Low |  |
|  |  | Create a behavior/gesture generator program | High |  |
|  | Robot Construction | Finalize design on robot body | Medium |  |
|  |  | Build robot body (torso, neck, head/face, arm, base) | High |  |
|  |  | Build mounts for Kinect, motherboard, HDD, etc. | High |  |
|  |  | Design and build power supply system | Medium |  |
|  |  | Controllers for servos | Medium |  |
|  | User Interface | Design and program user interface for input | High |  |
|  |  | Design and program user interface for robot face, status, and facial expressions | High |  |
|  |  | Record voices | Low |  |
| \*\*\*\*\*\*\*\*\*\* | Spring 2012 Deliverables | Interface with mobile base, improved mobile base navigation, some interaction with simple gestures and behaviors, torso and lower body are built. | \*\*\*\*\*\*\*\*\*\* |  |
| Summer 2012 | Body Construction | Complete arms | Medium |  |
|  |  | Complete head and neck | Medium |  |
|  |  | Test mechanisms of individual components | Medium | Motor control, range of motion |
|  |  | Assemble body to mobile base | Low |  |
|  |  | Assemble arms and head to torso | Low |  |
|  |  | Test completed robot body | Medium | Test each degree of freedom, navigation |
|  | User Interface | Improve interaction workflow | Medium |  |
|  |  | Improve UI graphics | Low |  |
|  | Vision | Object detection and recognition | High |  |
|  | Navigation | Fine-tuning navigation algorithms | Medium |  |
| \*\*\*\*\*\*\*\*\*\* | Summer 2012 Deliverables | Robot body completely built, robot able to navigate safely and reliably through the hallways, working user interface on screen | \*\*\*\*\*\*\*\*\*\* |  |
| Fall 2012 | Behavior | Improve behavior/gesture generator program | High | Include Forward/Inverse Kinematics |
|  |  | Add ability to go through door and use elevator | High |  |
|  |  | Connect with the semantic web database | Medium |  |
|  | Navigation | Implement automatic mapping algorithm | High |  |
|  |  | Find shortest route | Medium |  |
|  | Body Construction | Improve body construction if necessary | Medium |  |
|  | NLP | Improve grammar and vocabulary database | Medium |  |
|  |  | Improve understanding of sentence meaning and context | High |  |
|  |  | Add jokes to conversation | Medium |  |
|  | Miscellaneous | Interface with PAVE website for remote access | Low |  |
| \*\*\*\*\*\*\*\*\*\* | Fall 2012 Deliverables | Robot exhibiting more complex, interesting behavior and conversation, remote access via the PAVE website interface, better reliability of robot body construction, first public demo | \*\*\*\*\*\*\*\*\*\* |  |
| Winter 2013 | TBA | Buffer time, finish incomplete projects |  |  |
| Spring 2013 | TBA | Improvement and fine-tuning of the robot, test different algorithms |  |  |
|  |  |  |  |  |

# References

[1] Bhutada, A. (2011) *Universal Event & Motion Editor for Robots' Theatre*, master’s thesis (M.S.), Portland State University.

[2] Labunsky, D. (2010) *Facial Emotion Recognition System Based on Principle Component Analysis and Neural Networks*, master’s thesis (M.S.), Portland State University.

[3] Sunardi, M. (2010) *Expressive Motion Synthesis for Robot Actors in a Robot Theatre*, master’s thesis (M.S.), Portland State University.

[4] C. Breazeal, A. Brooks, J. Gray, M. Hancher, J. McBean, W.D. Stiehl, and J. Strickon (2003). "Interactive Robot Theatre", Communications of the ACM, 46(7), pp. 76-85.

[5] Durell Bouchard, Norman I. Badler: Semantic Segmentation of Motion Capture Using Laban Movement Analysis. IVA 2007: 37-44.

[6] Sidharth Dhawan, Arushi Raghuvanshi, Yale Fan, Frank Zhao, Yushi Wang and Marek Perkowski, “What’s that? Schrödinger Cat*!” Proc. the Third International Conference on Science and Technology, Policy, Culture and Communication,*  Korea, December 4-5, 2008.

[7] Arushi Raghuvanshi, Marek Perkowski, “Using Fuzzy Quantum Logic to learn facial gestures of a Schrödinger Cat puppet for robot theatre,” *Proc. ULSI 2008*, Dallas, May 2008.

[8] Quay Williams, Scott Bogner, Michael Kelley, Carolina Castillo, Martin Lukac, Dong Hwa Kim, Jeff Allen, Mathias Sunardi, Sazzad Hossain, and Marek Perkowski, “An Emotional Mimicking Humanoid Biped Robot and its Quantum Control Based on the Constraint Satisfaction Model,” *Proceedings of ULSI 2007.* May 13, 2007, Oslo, Norway.

[9] Arushi Raghuvanshi, Yale Fan, Michal Woyke, Marek Perkowski, “Quantum Robots for Teenagers,” *Proceedings of ISMVL 2007.* May 14-15, 2007, Oslo, Norway.

[10] Dong Hwa Kim, Christopher Brawn, Maciej Sajkowski, Tomasz Stenzel, Tsutomu Sasao, Jeff Allen, Martin Lukac, Tzewen Wang, and Marek Perkowski, “Artificial Immune-Neuro-Fuzzy System to control a walking robot Hexor, *Proc. ULSI* 2006.

[11] Christopher Brawn, Natalie Metzger, Jake Biamonte, Martin Lukac, Akashdeep Aulakh, Indudhar Devanath, Maciej Sajkowski, Tomasz Stenzel, Dong Hwa Kim, Tsutomu Sasao, Marek Perkowski, “Hexor, a Walking and Talking Robot with Quantum and Fuzzy Inference,” *Proc. ULSI* *2006,* *Workshop on Post-Binary ULSI Systems*.

[12] Phil Serchuk, Ehud Sharlin, Martin Lukac, and Marek Perkowski, The Vagueness of Robot Emotions, *Proc. ULSI 2006, Workshop on Post-Binary ULSI Systems.*

[13] Marek Perkowski, Tsutomu Sasao, Jong-Hwan Kim, Martin Lukac, Jeff Allen, and Stefan Gebauer, “Hahoe KAIST Robot Theatre: Learning Rules of Interactive Robot Behavior as a Multi-Valued Logic Synthesis Problem,” *Proceedings of the 35th International Symposium on Multiple-Valued Logic,* May 19 - May 21, 2005 at Calgary, Canada, pp. 236-248.

[14] Phil Serchuk, Ehud Sharlin, Martin Lukac, and Marek Perkowski, “Emotion-mapped Robotic Facial Expressions based on Philosophical Theories of Vagueness,” *ACM Conference on Human Factors in Computing Systems (CHI 2006) Workshop on HCI and the Face*, April 22-27, 2006, Montréal, Canada

[15] Evan Ackerman, Video: Kombusto, MIT's Interactive Dragon Robot, Internet article, URL: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/video-kombusto-mit-interactive-dragon-robot

# Appendix A

Specifications on the Zotac Motherboard IONITX-L Series

Motherboard Specifications

**Chipset**

NVIDIA® MCP7A-ION Series

**Size**

Mini-ITX form factor of 6.69 inch x 6.69 inch (171mm x 171mm)

**Microprocessor support**

Intel® ATOMTM 230/330 CPU

Support for 533 MHz FSB

**Operating systems:**

Supports Windows XP 32 bit/64 bit, Windows Vista 32 bit/64 bit and Windows 7 32 bit/64 bit

**System Memory support**

 Supports dual-channel (128 bits wide) DDRIII memory interface

Supports DDRIII 1066/800

Maximum memory size: 4 GB

**USB 2.0 Ports**

Supports hot plug and play

Ten USB 2.0 ports (six on the back panel, four via the USB brackets connected to the internal USB headers)

Supports USB 2.0 protocol up to 480 Mbps transmission rate

**Onboard Serial ATA**

Independent DMA operation on four ports (three onboard SATA headers, one rear panel e-SATA).

Data transfer rates of 3 Gb/s.

**On board RTL8211CL Gigabit LAN (Optional)**

Supports 10/100/1000 Mbps operation

**On board RTL8201EL Fast Ethernet** (**Optional**)

Supports 10/100 Mbps operation

Supports half/full duplex operation

**Onboard Audio(Optional)**

Azalia High-Definition audio

Supports 6-channel

Supports Jack-Sensing function

**Green Function**

Supports ACPI (Advanced Configuration and Power Interface)

RTC timer to power-on the system

AC power failure recovery

**PCIE x16 Interface**

PCIE x16 Generation 2.0 compatible

 5 GHz support, for a total bandwidth of 5 Gbps per direction per lane

 Wake up function is supported

 Clock spread spectrum capability.

**Onboard Graphics support**

 Integrated 300 MHz DAC for analog displays with resolutions up to 1920x1440 at 75 Hz.

 Integrated GeForce 9xxx Series GPU, Supports DX10

 VGA/DVI/HDMI output support (optional)

**Integrated HDMI Interface with HDCP**

 Support Dual link DVI,resolutions up to 2560x1600

 Supports DVI or HDMI 1.3 interfaces

 Secure digital audio merged from integrated HDA codec with no external audio signals required

 Support for HDCP 1.3 using soft or hard HDCP keys

 HDCP encryption support when configured as DVI or HDMI link without the need for external HDCP key crypto ROM

**Dual Head Display Controller**

 Full NVIDIA nViewTM multi-display technology capability, with independent display controllers for the CRT, TMDS, DisplayPort, and HDMI interface

 Each controller can drive same or different display contents to different resolu- tions and refresh rates

**Expansion Slots**

 One Mini PCI Express slot

 One PCI Express x16 slot

1. An article covering the Robots play: http://www.swissinfo.ch/eng/culture/Robots\_steal\_the\_musical\_show.html?cid=996590 [↑](#footnote-ref-1)
2. An article covering the Hataraku Watashi play: http://news.bbc.co.uk/2/hi/7749932.stm [↑](#footnote-ref-2)
3. OpenKinect web page: http://openkinect.org/wiki/Main\_Page [↑](#footnote-ref-3)
4. Source: http://www.scootercatalog.com/ev-rider-stand-n-ride-electric-scooter.html [↑](#footnote-ref-4)
5. Source: https://www.msu.edu/~luckie/segway/rmp/rmp.html [↑](#footnote-ref-5)