Collaborator Kickoff Document

Description

The end-product will collaboratively operate on a shared workspace with human workers. What is meant by “collaboration” is that the robotic arms are currently unable to properly detect humans as cooperative agents. Our application will provide this core functionality. In detail, collaboration is a means to decrease workload of human workers by operating in the same environment through task sharing. The end users of the product will be those who lack human resources on labor-intense jobs on production lines.

Master Feature List

- MF-1: The robotic arm will be able to simply plan its motions for any valid 3D pose in the workspace.
- MF-2: The arm will be able to avoid colliding with uniform and simple obstacles in the workspace.
- MF-3: The arm will be able to avoid colliding with any form and size of obstacle in the workspace.
- MF-4: The arm will be able to recognize human-alike models to distinguish an ordinary object with a cooperative agent.
- MF-5: The arm will be able to recognize models that describe human and non-human in more detail such as cylinders for both arms and legs instead of cylinder of a human size.
- MF-6: The arm will be able to evade from human models without disrupting its own functionality in an unlimited 3D space, i.e. still realizes its job, but might be in a longer time span. This is a trade-off for its reliability.
- MF-7: The arm will be able to avoid from human models without disrupting its own functionality in a restricted 3D space, that is shared region.
- MF-8: The arm will be able to understand the hand gestures of human workers that indicate the human workers are done.
- MF-9: Prioritization of human worker based on collision probability measured by distance.
- MF-10: Avodiance of the robotic arm in the shared workspace when there is risk of collision with human worker.
- MF-11: The arm will be able to recognize whether the case is full or not. When full, it will automatically start its own operation.
- MF-12: The arm will be able to stack the cases referred as in the sample scenario.
- MF-13: The arm will have a safety layer that filters the physically damagable commands.
- MF-14: The arm will be able to operate in an infinite loop as embedded systems dedicated to accomplish its tasks.

Bonus Features:
• The robot will be able to operate with multiple workers.
• The robot will be able to detect human worker(s) out of shared region.
• The robot will be able recognize voice commands of worker(s).

Work Packages

WP #	Term	WP title	Estimated number of man-months
1	491	Experimental Setup	3
2	491	Motion Planning	6
3	491	Perception	6
4	491	Simulation	-
5	492	Integration	3
6	492	Safety	6
7	492	Testing**	6

** Testing will be an interleaved work package that will be done throughout all work packages.

Detailed Descriptions of High-Level Workpackages

WP-1 Experimental Setup

Experimental Setup includes the integration of Kinect to Ubuntu and ROS, as well as the environment modeling in simulation and in physical world. Environment modelling is, in detail, the designation of a simple table, initially an empty box, then a full box on it which is based on the scenario proposed and after all robotic arm. This work package does not include any Perception or Motion Planning, but our goal in here is to devise an environment in which the robot arm can complete its tasks such as we proposed in our sample scenario. No MFs will be completed at the end of this work package, however the implementation of MF1 requires this WP to be satisfied.

WP-2 Motion Planning

In realistic environments, robot agents usually perform their tasks by following a finite number of steps. The tasks of robot agents usually involve motion such as moving from an *initial* point to another *goal* point where the robot may initiate another sequence of steps to achieve another task. However, as human beings do in their real life, a robot will not be able

to move effectively without a plan. Effectiveness of motion in this situation is measured by how focused the agents actions towards achiveving its goal. In order for the robot to reach its desired location effectively, it needs a proper plan. In Robotics literature, this concept is called **Motion Planning**. Before starting to move, the agent will construct a path considering what entities are in the environment that affects the shape, length of the resultant path and the complexity of construction. These entities are generally called *obstacles*. As one can predict, an environment can have both static and dynamic obstacles with static ones stand still and dynamic ones move continuously and/or periodically. The agent will execute an *algorithm* that will construct a path by connecting intermediate points (or locations) in the environment without causing any collisions with neither static not dynamic obstacles; otherwise, physical and economical damage may happen. For the sake of safety, Motion Planning process of a robot agent should perform effectively (i.e. avoiding from physical and economical damage). As to the efficiency of motion tasks, the process should execute efficiently as well. This will make a good contribution to the agents effectiveness by reducing possibility of entering unnecessary, and possibly risky, situations.
This module play a significant role in the end-product with affecting the safety issues and efficiency in scenarios such as production lines where an agent must make less mistakes to avoid from possible economical loss. Since applying the module into the real environment and scenarios is difficult without proper generalization of the whole problem of planning, most of the time spent for development of this module will be in a simulation environment, **Gazebo** and/or **RViz**. The flow of development is following:

- Firstly, the robot agent, in this project’s case the agent is a robotic arm, will perform motion planning in a *collision-free* environment to grasp the approach to solve planning problem provided by the motion planning library, **MoveIt**.
- Secondly, motion planning will be performed in a collision-rich environment, where only static obstacles will exist, to test various approaches of the library if the robot will move in optimal, if not almost optimal, paths to the goal.
- Thirdly, dynamic obstacles, for example a rigid body moves periodically, will be introduced to the environment, in which the robot shall perform motion planning.

As to the milestones of this project, the first milestone is associated with this module involving a basic scenario. In this scenario, the robot arm will have information of the environment’s boundaries, workspace boundaries, the number of static obstacles with their position and orientation. The robot will have this information provided by a computer that holds a control module to send proper commands to start and stop the planning process and by using the given specifications of workspace the robot will perform motion planning to target objects, *boxes*, to perform **pick** action and perform planning again to **place** them to a determined location. If this target location is same for some or all of the boxes, the robot arm will *stack* them. While performing, the robot will have no risk of interference from a dynamic obstacle, especially a human, and no supervision provided by *Perception* module (next section).
As for master features of the end-product, **MF-1**, **MF-2** and **MF-3** will be directly involved and aimed to be working properly, whereas **MF-6**, **MF-7**, **MF-9**, **MF-10** and **MF-12** will be partly involved since they require Perception module to be ready.

WP-3 Perception
Perception is the organization, identification, and interpretation of sensory information in order to represent and understand the presented information, or the environment.
This work package is aiming to assist MF-4, MF-5, MF-7, MF-8,MF-11 by adding our robot vision skills.

As already denoted in Motion Planning work package, relevant master features will be accomplished in cooperation with this work package. Specificially, the perception of different models will be handled incrementally such as the box, ingredients of the box, the robot, the table and eventually human. When this tasks are properly accomplished, gesture recognition is going to be implemented. In the early phase of the work package, we will represent the shared workspace with a markup in the simulation, such as distinct RGBD (D stands for Depth) values read by Kinect2 sensor. Then, the program and the robotic arm will understand the region constitutes the shared volume. We can also use a characteristic vision-band to distinguish the shared volume. Also as a bonus feature, the speech recognition can be implemented in a way that recognizes the HALT orders of the human worker(s).

WP-4 Safety
Safety, in the sense of our project, considers human involvement as the key criteria. Human involvement means human worker's recognition in technicality. This work package comprises two different safety layers, which are software and hardware.The development of MF-13, which implements software-side safety policies, will be done in this work package. Also, the hardware-related safety policies will be handled in the lab during this work package. Since human worker is thoroughly different than other models and has a much more sensitivity to the damages, we need to devise safety measures in Motion Planning package along with Safety package. Also, the noisy inputs produced in Perception work-package must be purified in here to eliminate the possibility of ambiguity.

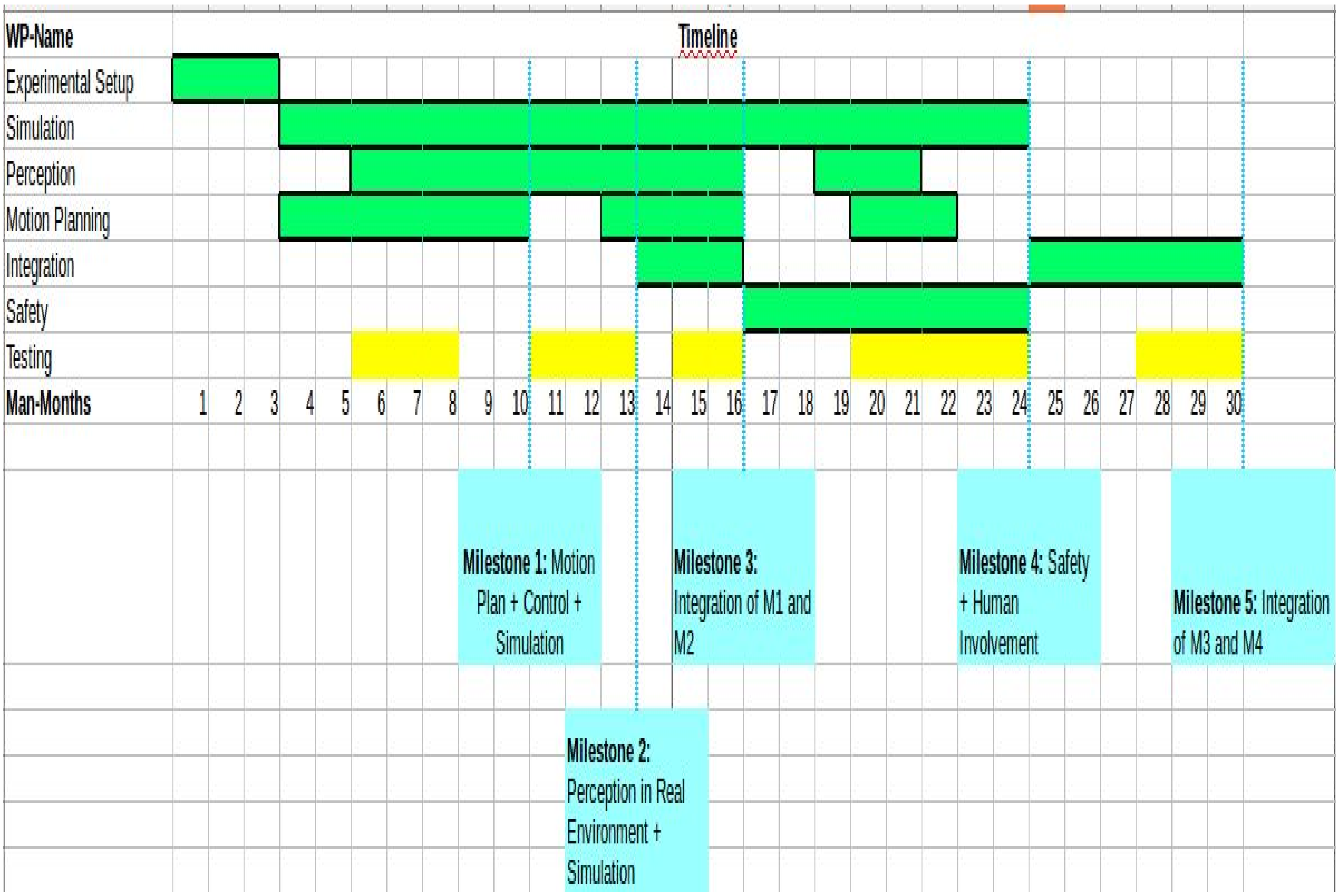
WP-5 Integration
This work package will cover the integration of other master features periodically, finally to fulfill MF-14.Firstly, Perception and Motion Planning work packages will be integrated. Later, integration of Safety work package is aimed to be added to the project.

WP-6 Simulation
The simulation environments we will be using are Gazebo and Rviz. The Simulation work-package will be interleaved one, since we will be dealing with robotic arm and milestone scenarios in the virtual environment most of the time and then we will move to physical environment when all testing and simulation produce passing results. Also, this work package includes the early phases in all other work packages since it is needed to prepare virtual environments.

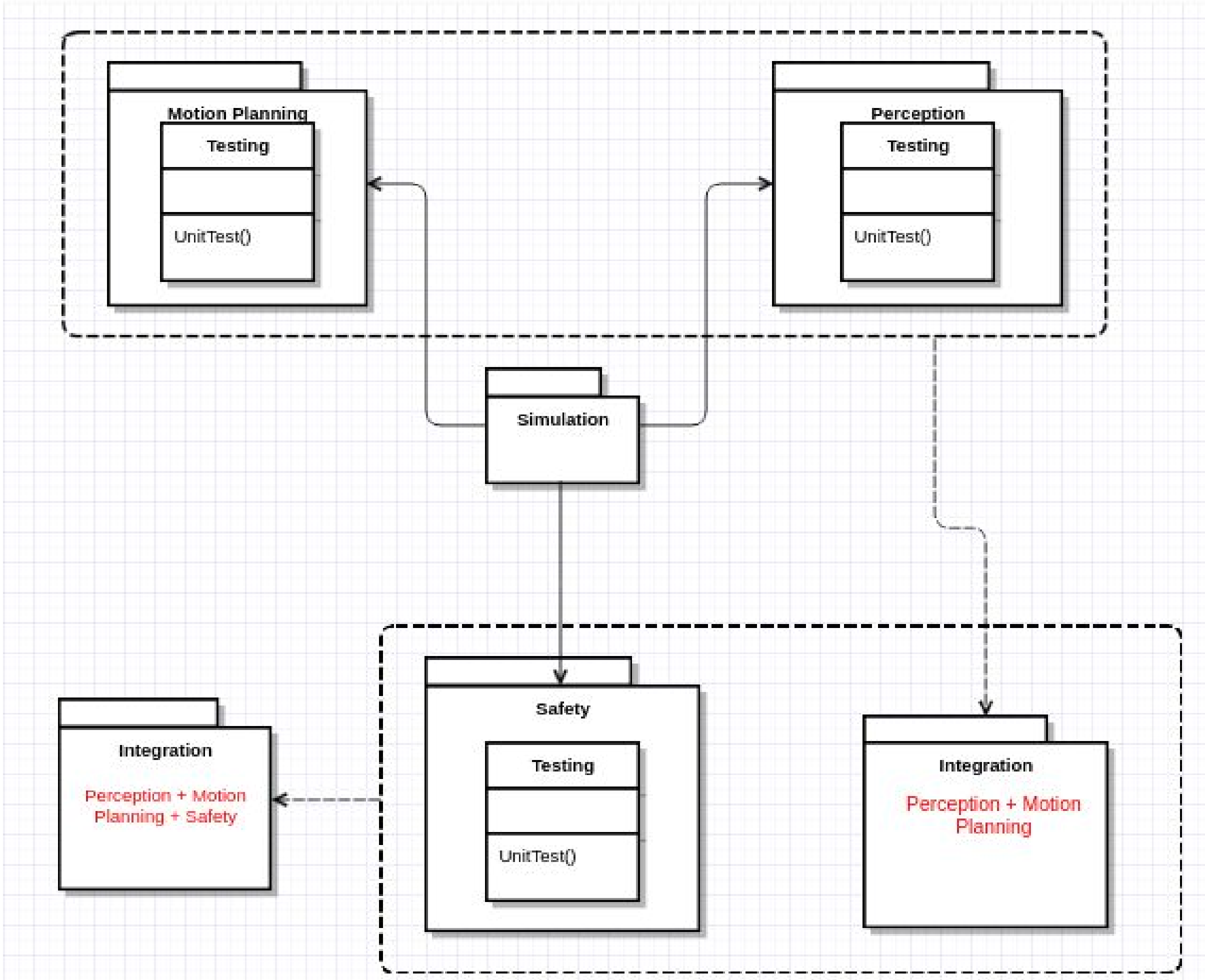
WP-7 Testing
In ROS, there is a Unit testing utility, which we will hugely exploit in our all development process and work packages. Also, mostly in Integration work package

we will do component testings to make sure that the integrated parts work properly. An example to unit testing specific to our project is the testing of the data exchange between the Perception module and the Motion Planning module. These data are basically transmitted through ROS topics and all will have distinct ROS message types and the unit testing process will provide us reliability on these primal features.

Timeline



Overall Systems Architecture



Risk Assesment

Risk #	Description	Possible Solution(s)
1	Noisy image output of the Perception Module.	Reduction of the complexity of recognized human model.
2	The improper constraints of the Motion Planning Algorithms that are to be used.	Hybrid algorithms which cancels each other drawbacks.
3	[-Pi,Pi] restriction of MoveIt Framework in order to avoid from possible singularity orientations.	Introduction of quaternions to the MoveIt API.