CHARM: A Platform for Algorithmic Robotics Education & Research

Surya P. N. Singh[1], Hanna Kurniawati[2], Kianoosh Soltani Naveh[3], Joshua Song[4], and Tyson Zastrow[5]

***Abstract*— This paper introduces autonomous sorting of moving coins via a robot as a multi-level task that supports the principled study of robotics fundamentals including kinematics, dynamics, perception, motion planning, controls, and optimization based around a widely obtainable, standardized, low-cost object (a coin). The paper also presents a demonstrated solution to this in the form of the CHARM (Coin Handling Arm for Robotics Mastery) robot, which addresses the autonomous coin sorting problem using an economical kit made from commodity computing hardware and three Dynamixel servomotors. From a learning perspective, this problem facilitates interdisciplinary practice across subject and grade levels with an algorithmic foundation that is central to modern robotics. Evidence supporting this approach is illustrated from case studies of student projects and, in particular, the CHARM robot. Beyond practice alone, by presenting a challenging (but manageable) research problem, we found that the coin sorting task teaches robotics in a principled way. Further, algorithmic complexity tiers the problem to academic levels. While motivated by robotics education, the (optimal) coin sorting problem may also be seen as an archetype problem for manipulation/motion-planning research. Thus, this also promotes a research foundation supporting later research opportunities.**

I. INTRODUCTION

Not only is robotics an increasingly prevalent and an important part of everyday life that captivates the imagination, but it is an inherently fascinating subject. From US FIRST to lunar rovers to new graduate programs in robotics, interest in the subject has increased both popularly and academically. This, in turn, has renewed interest in introductory robotics courses, particularly at the mezzanine level.

From a learning perspective such courses offer an opportunity to introduce systems engineering concepts and to integrate knowledge across multiple disciplines and topics. From a teaching perspective, these courses attract highly motived and engaged students due to the general enthusiasm for the subject. While such excitement is helpful, the applied nature and general expectation of robotics often implies interest in new material and "modern" results. Compared to the significant attention paid to curriculum and learning development within particular subdisciplines, the design and emphasis for (introductory) robotics courses as a gestalt has received less attention [1], [2]. In part, this is attributable to the interdisciplinary and expanding nature of robotics, which has grown from articulated serial kinematics chains to mobile systems with integrated sensing and control. Both the breadth of material and the (relatively) short course periods suggest the need for a careful structuring of such courses [3].

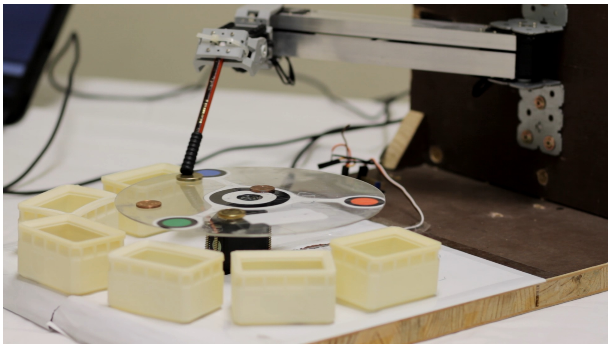

Fig. 1. The CHARM robot autonomously sorts coins randomly placed on its turnable. The robot involves several algorithmic principles including object recognition, kinematics/control, and motion planning in dynamic environments

Robotics laboratories spans the gamut from courses that use robotics to support course topics to the entire art of robotics. Thus, many robotics kits are focused on teaching closely related subjects such as programming [4], dynamics and controls [5], [6], and mechatronics [7] more so than to the principles of robotics. Hence there is a need for laboratory designs with a focus on algorithmic principles that enables students to navigate robotics research results and to apply these methods. We delineated the algorithmic areas in robotics using the Robotics and Automation Society's program structure as a guide. From this we identified algorithmic areas as motion planning, perception, kinematics, mapping, machine learning, control, and systems.

The problem of autonomously sorting moving coins (see Fig. 1) addresses the above impetus in a step-wise manner that allows teams of students to study these principles at increasing levels of complexity so as to match their level of learning and engagement [8], [9]. While coin sorting mechanisms abound, the sub-tasks involved present several challenges including vision [10] and control [11], especially for autonomous operations. More generally, the coin sorting task has subproblems that involve each of the aforementioned algorithmic areas – end-effector position placement, velocity control (Jacobeans), object recognition, obstacle avoidance, system identification, autonomous operation, and underactuated (saturated) controls.

This work was supported by the Mechatronics Program and the Robotics Design Lab (RDL) at The University of Queensland (UQ), Brisbane, Queensland 4072, Australia

[1] Surya P. N. Singh (corresponding author) is RDL and Mechatronic Program director and coordinates METR4202 (Advanced Controls and Robotics), `spns@uq.edu.au`

[2] Hanna Kurniawati is with the RDL and coordinates COMP3702 (Artificial Intelligence), `hannakur@uq.edu.au`

[3] Kianoosh Soltani Naveh is a Mechatronic Engineering student in the UQ RDL, `k.soltaninaveh@uq.edu.au`

[4] Joshua Song is a Mechatronic Engineering student in the UQ RDL, `mun.song@uqconnect.edu.au`

[5] Tyson Zastrow is a Mechatronic Engineering graduate, `tysonzastrow@gmail.com`

In this paper we also introduce a candidate system in the form of the CHARM (Coin Handling Arm for Robotics Mastery) robot (pictured in Fig. 1). It is a robotic platform with a camera, robot arm and a workstation that allows students to explore different topics of robotics and mechatronics engineering such as computer vision, motion planning and mechanics from elementary to advanced. An example with novel motion planning, perception and kinematics is demonstrated to highlight its educational capabilities.

The remainder of this paper is structured as follows. In Sec. II, we detail the autonomous dynamic coin sorting problem and show that it is, from an algorithmic perspective, at least P-space hard. The design of the CHARM robot and how it addresses these algorithmic aspects is detailed in Sec. III. A discussion of this from algorithmic robotics education perspective is in Sec. IV.

II. THE COIN SORTING PROBLEM

The basic coin sorting problem is designed for mezzanine (i.e., upper undergraduate and beginning graduate) students. It is about moving coins to an appropriate bin, in the sense that all coins in the same bin must be of the same type. We add a twist to the problem to make it more interesting: the coins are placed on a rotating table (to which there can also be other objects/obstacles placed around it). Let us assume we have n coins of k different types. Suppose each coin is placed on top of a turning table that rotates with an angular velocity (which is typically, but not necessarily, constant) and suppose k empty bins are placed right outside of the turning table (see Fig. 2). The goal is to move each coin from its initial position to one of the bins, such that all coins inside a bin belong to the same type. The problem ends when no more coins are on the rotating table. The problem is solved when it ends with all bins having the same type of coins. To solve this problem, students were given three low-cost motors and a camera.

This coin sorting problem spurs study in at least four robotics areas:

1) **Hardware design.** At the very least, students must be able to design and build a simple 3-DOF arm that can push a coin. In addition, the basic coin sorting problem teaches the trade-off between hardware design and algorithmic difficulty. For instance, to solve the problem, one can build an arbitrary shaker, which is easy to built. However, such a shaker would make the process error prone and stochastic, especially when the shaker is controlled by imprecise, saturated motors (as is sometimes the case with low-cost motors).
2) **Perception.** The system must be able to identify, differentiate, and tracks multiple moving objects. While coins might be a standard object, variations in lighting and coin wear/quality additionally highlight (and teach) the challenges in developing robust sensing techniques. Since the types of coins can be increased easily, the coin sorting problem also teaches the challenges in scaling various vision and/or object classification techniques.
3) **Motion Planning.** The motion planning component of the coin sorting problem can be defined more generally as follows. Given n closed and planar objects on a 2D Euclidean plane, where all objects orbit with the same constant angular velocity around a common barycenter. Let $g_i \subseteq \mathbb{R}^2$ be the goal region of object-i for $i \in [1, n]$, and let each goal region acts as a sink, in the sense no object can leave the region. Suppose the initial position of each object is known and is within a finite distance from the barycenter. Then, we want to find a continuous path that moves each object-i from its initial position to its goal region g_i ($i \in [1, n]$) without colliding with any other object. This problem is not just complex, it is at least PSPACE hard, as the problem of motion planning with movable obstacles where the obstacles must end at a pre-specified location – a PSPACE hard problem – is a special case of the above problem [12].
4) **Estimation.** The use of low-cost motors require students to be able to estimate and filter out errors when controlling the robot. Furthermore, to perform well, students must also estimate the angular speed of the turning table. Although this estimation can be done easily by placing markers on the turn table, the ability to identify the need for such estimation and the use of such markers are skills that could come handy in developing more complicated robotics system.

The aforementioned list is not fixed. Depending on the systems design, some problems may be more prominent than others, and additional problems may occur. Furthermore, aside from varying levels of difficulty in each sub-field, the coin sorting problem also provides the opportunity to study the interplay between various aforementioned components, such as between perception and motion-planning. That is, for example, certain camera (sensor) placements may reduce occlusions (simplifying perception), but may not allow as encompassing a view (complicating planning/control).

The coin sorting problem can be extended or simplified easily to better cater to various student and/or learning requirements. For instance, it could be simplified by having a stationary table and direct sensing of the target and its location (e.g., augmenting the coin/object with a magnet for easier detection and/or to incorporate switches, encoders, or inductive sensors under the table to simplify object localization). Control performance can be simplified, for example, by providing larger bins (relative the the target size).

Similarly, the problem may be adapted to be more challenging or rigorous by considering variations of (algorithmic) complexity. For instance:

- Adding 2.5D obstacles that moves on the rotating table – This adds significant difficulty in perception, planning, and estimation.
- Introducing adversity to the system (e.g., an opponent that competes for the coins) – This would allow for the incorporation of game theory and/or AI strategies.
- Having a time optimal solution – This is still an open research problem.

III. CHARM ROBOT

The CHARM (Coin Handling Arm for Robotics Mastery) robot is a coin sorting system whose novel feature is its ability to slide coins on a turntable without ever colliding other coins on the table. It is a candidate solution to the Coing Sorting problem.

Its design demonstrates the features of an algorithmic approach to in robotics laboratories. CHARMs different hardware parts are shown in Fig. 2. The robot uses a camera mounted on the top overlooking the workspace. The workspace includes a spinning table with a number of different coins to be sorted. The spinning table features a marker made of three circles red, blue and green respectively. The robot arm is made of three Dynamixel servomotors (Robotis, Seoul, Korea) and is mounted to the side. This arm touches the table using an extension that allows the arm to slide the coins.

The arm is controlled by software on a PC. The software which is written in C++ can be divided to three main modules:

1) Detection and Classification,
2) Motion planning, and
3) Kinematics.

The detection and classification module uses OpenCV to process image frames obtained from the camera and identify and locate coins. The motion planning and kinematics module uses the findings of detection module to plan the trajectory necessary to slide the coin to its respective bin without colliding with the rest of the coins.

A. Robot Arm Kinematics and Dynamics

The robot arm consists of three Dynamixel servos at each joint giving an RRR arm. Kinematically, the robot arm is positioned in the plane of turn table. The third joint then allows the arm to touch the coins and slide them (touch down motion). For this reason, the kinematics analysis of planar motion and touch down motion can be carried out separately. In the kinematics analysis the coordinates were chosen such that $x - y$ represent the turn table coordinate and z the normal axis as shown in Fig. 3. From Fig. 4, the $x - y$ plane forward kinematics equations are equivalent to that of an RR arm:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \tag{1}$$

The inverse kinematics equations are derived using geometrical properties of triangle:

$$r^2 = L_1^2 + L_2^2 + 2L_1L_2\cos\theta_2 \tag{2}$$

$$\Rightarrow \theta_2 = acos\left(\frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \tag{3}$$

$$\theta_1 + \alpha = atan\left(\frac{y}{x}\right) \tag{4}$$

$$\sin\alpha = \frac{L_2 \sin\theta_2}{r} \tag{5}$$

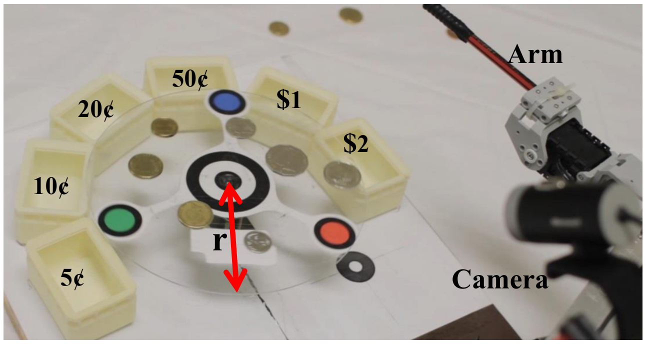

Fig. 2. Overall system layout of the CHARM robot showing the coins and turntable radius (r) relative to bins and the camera and arm locations

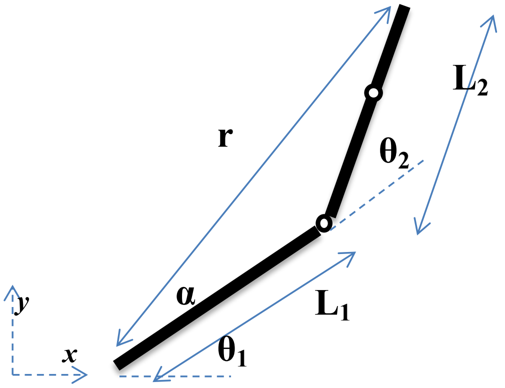

Fig. 3. Top view of the CHARM kinematics

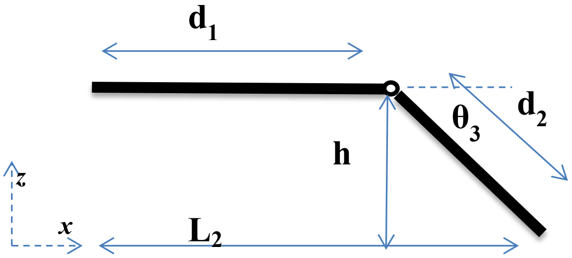

Fig. 4. Side (touch down) view of the CHARM kinematics

The touch down motion is along the z axis (see Fig. 4). This motion requires the third joint to rotate a specific angle so that the tip of the arm touches the surface of the table. As this joint bends the tip of the arm, the radial distance of the tip is affected. This is shown in the following analysis:

$$h = d_2 \sin\theta_3 \tag{6}$$

$$\Rightarrow \theta_3 = asin\left(\frac{h}{d_2}\right) \tag{7}$$

$$L_2 = d_1 + d_2 \cos\theta_3 \tag{8}$$

In addition to positioning the arm and reaching the coins, velocity control was critical to the correct operation of the robot, as it would need to be able to intercept the moving targets (i.e., the coins on turning table). To implement the velocity control of the robot arm a trajectory with time stamps at each point is generated by kinematics module and each point is sent as a position command to the servo. This method allowed an implicit setting of the velocity while minimizing the communication between PC and servos.

B. Dynamics of the Turn Table

The turn table consists of a disk on a continuous rotation Plolu servo. The turn table's speed can be controlled via a GUI based program on PC. It is also possible to write programs to control the speed of the turn table if necessary. The turn table forms the main part of the workspace with the option to put targets or objects on the surface and use the arm to manipulate those object. It also features special markers that allow the camera detect the position, distance and orientation of reference frame of the workspace. More details are given in the next section.

C. Perception

The detection system uses a Lifecam Studio (Microsoft, Redmond, Washington) in collaboration with the OpenCV toolkit. The camera is mounted above the arm facing the rotating turntable; this gives an oblique view of the table surface which helps to avoid erroneous detection of spherical or cylindrical objects which can be mistaken for coins when viewing the table from directly above.

The coin detection algorithm uses edge detection and ellipse fitting to detect elliptical shaped objects on the table surface. Firstly the image is undistorted using pre-calculated camera calibration parameters, and then a Canny edge detector is used to find all edges in the scene. These edges are subsequently turned into OpenCV contours (i.e., vectorized edges), and then broken into individual curve segments based on the change in curvature of the contours [13].

Then an ellipse is fitted to each curve based on the points in that curve and the fitting error is calculated, any ellipses with high errors are discarded. The remaining ellipses are then consolidated by searching for and removing duplicates (i.e., ellipses with very similar parameters).

The next step is identifying the turntable, this is done via concentric circles printed on the table. Circular markers were chosen as they can be detected using the same technique as finding coins. The list of ellipses is searched for concentric ellipses with relative sizes that match that of the centre table marker, once that is found the three directional markers are then identified using a similar concentric search with the addition of colour sampling at the centre of the ellipse. Once the turntables position relative to the camera is established a mask can be employed to ignore any detected ellipses outside the tables surface as well as ignoring ellipses that have a skew different to that of the table surface. Any remaining ellipses must be coins (or coin shaped objects) on the table.

The detected coins are then classified based on colour and size. Colour classification uses a histogram of the coins pixels in the YCbCr colour space and compares this to a training set histogram for both gold and silver coins. The match error for gold and silver is calculated using the Chi-Square method provided by OpenCV and recorded. The real world size of the ellipse is calculated using the distance to the table (calculated from the tables markers) and the focal length of the camera. This size is then compared to every actual Australian coin size and the error squared is recorded for each. The ellipses position relative to the rotating table is then calculated and stored with all the calculated error values.

This entire process is repeated for an arbitrary number of frames, when new ellipses are detected their table position is compared to previous ellipse table position to see if it is the same coin as previous detected, if it is then its error values can be collated with previous matches to that coin to provide more data for classification. In this way classification can be improved by allowing more frames (i.e., more data) to be collected.

When sufficient frames have been captured (we used 20) classification of the detected coins takes place; first ellipses that have been detected in less than half the captured frames are discarded as they are unlikely to be actual coins, then the value of the remain coins are determined by finding the coin type with the minimum combined colour and size error. We found size to be a more accurately distinguishable feature and as a result weighted its error values 3 times higher than colour errors. Also note that at this point ellipses with minimum match errors above a certain threshold were also discarded as the confidence of the match was simply too low.

Figure 5 shows the results of the coin detection and classification.

```
TOTAL COIN VALUE: $12.20
( 2x5c 2x10c 2x20c 1x50c 1x$1 5x$2 )
Coin 0: 50c (conf: 0.065109, count: 16, dia: 32.49, loc:[5.1, -62.8, 1.2])
Coin 1: 5c (conf: 0.178546, count: 19, dia: 19.63, loc:[-3.8, 61.1, 1.5])
Coin 2: $2 (conf: 0.099546, count: 15, dia: 20.99, loc:[36.6, -76.8, 3.7])
Coin 3: 10c (conf: 0.172230, count: 20, dia: 24.00, loc:[38.7, -46.8, 1.2])
Coin 4: $2 (conf: 0.202903, count: 18, dia: 20.63, loc:[42.0, 35.2, 0.9])
Coin 5: $2 (conf: 0.116376, count: 20, dia: 21.02, loc:[39.2, 70.5, 2.6])
Coin 6: 5c (conf: 0.117539, count: 19, dia: 19.95, loc:[73.8, 24.9, 2.2])
Coin 7: 10c (conf: 0.098642, count: 19, dia: 24.05, loc:[-66.7, 4.5, 1.5])
Coin 8: 20c (conf: 0.106803, count: 19, dia: 29.12, loc:[-66.4, -31.0, 1.7])
Coin 9: 20c (conf: 0.044565, count: 13, dia: 28.88, loc:[54.1, -12.1, 0.7])
Coin 10: $1 (conf: 0.149936, count: 14, dia: 25.24, loc:[-49.9, 34.5, 1.0])
Coin 11: $2 (conf: 0.001598, count: 1, dia: 20.91, loc:[-41.7, -4.8, 2.6])
Coin 12: $2 (conf: 0.001769, count: 1, dia: 20.42, loc:[-39.0, -24.5, 2.1])

Table Speed: 2.2378 (avg: 2.0168)

Processing time: 0.773432
```

Fig. 5. Coin detection and classification

D. Motion Planning

The motion planning problem can be described as multi-goal motion planning in a known dynamic environment. Although the coins act as moving obstacles, their path and velocity can be estimated from the video camera input.

The first part of the problem is deciding the sequence in which coins are sorted. This is similar to the classic travelling salesman problem, except the possible paths between each coin and its respective bin vary over time. To solve this, a greedy approach was used where paths are found each coin to its bin, and one path (the shortest) is chosen. The future positions for the remaining coins is then estimated, and the process is repeated until all coins are sorted.

The second part problem is the path finding for getting a coin to the bin. The algorithm adopted was a grid-based A* search, with straight-line distance to the goal as the heuristic. This was chosen over probabilistic roadmap methods as it is less computationally intensive. The grid is 3-dimensional, consisting of x and y coordinates as well as time. The search tree begins at the coin's starting position and an estimated start time. A bidirectional search is not possible as the time of arrival at the goal is unknown until the path is found.

As each point is explored, the time at the point is estimated based on the path length divided by end-effector velocity. This time is then used to estimate the obstacle positions. A simulation result of the generated search tree is shown in Fig. 6. The points explored during path finding are shown as green dots, and the positions of the obstacles over time are shown as red circles.

Once the paths have been found, the planner sends the trajectory, which consists of time-stamped coordinates, to the arm controller which performs the inverse kinematics.

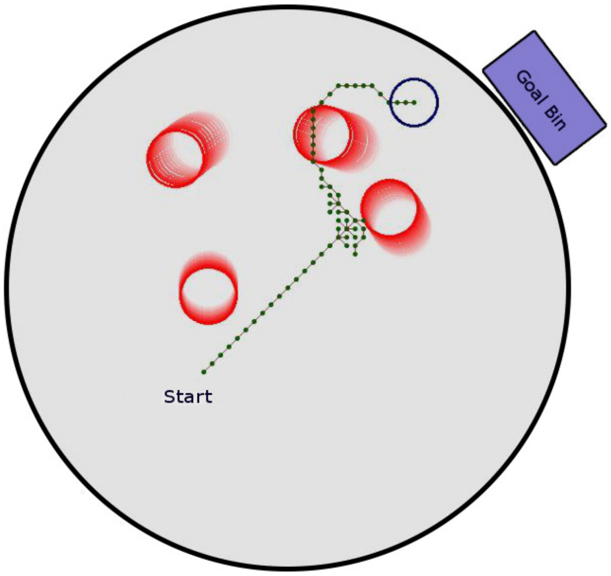

Fig. 6. A sample motion planning trajectory

IV. DISCUSSION – TOWARDS ALGORITHMIC ROBOTICS EDUCATION

As demonstrated by the CHARM robot, the coin sorting task presents several integrated challenges. While it is quite feasible to solve the problem using Lego kits (as was done by other student teams), the use of a similar (in price and number of degrees of freedom) kit based on three Dynamixels was more flexible, but is more complicated to develop. The course and project were assessed using course feedback surveys. Compared to the previous year the course was administered, the course showed a slight increase in how well the materials helped them with the learning (from 75% of students in agreement to 91% of students in agreement). Overall course marks are also high (4.81/5, with 100% of students rating the course as satisfactory, the strongest response for a class in the Mechatronics program).

A. Robotics to Support Related Studies

As a capstone subject, robotics integrates knowledge across multiple disciplines and topics. This highly positive and attractive characteristic makes it well suited to studying systems engineering. From a teaching perspective, robotics courses attract highly motived and engaged students due to the general enthusiasm for the subject.

Robots have been found effective in engaging and reinforcing student learning not only in robotics classes but also as a general learning tool to help students understand physical and mathematical concepts such as geometry and kinematics [7]. Therefore it has been an apparent choice for teaching robotics and mechatronics. In doing so, many instructors have used LEGO Mindstorm as the preferred robotics kit due to simplicity, reusability and ease of prototyping [14]. These kits have allowed instructors teach the concepts of direct and inverse kinematics and computation of simple arm trajectories. However limitations imposed by LEGO Mindstorm software and interface have also been reported. These limitations and the weakness of labs in teaching robotic algorithms such as vision and motion planning seem to correlate. Rosenblatt et *al.* [15] have reported, in designing lab assignments for a robotics course in Carnegie Mellon University, they used parts of LEGO Mindstorm kit in combination with their assorted sensors and controllers to let students come up with more creative solutions. However the labs do not provide a unified problem and therefore each requires different setup.

Other robotic kits such as Pendubot have allowed students implement theories in the labs on one setup, but due to the nature of the robots the topics become limited to specific areas such as control and systems [16]. On the other hand, mobile robots such as e-puc robot, Roomba and many others have tried to provide one platform for learning different robotics concepts in signal processing, control and distributed intelligent systems [4]. However, these applications do not form a well structured robotics problem and most of them do not support computer vision. Moreover, taking into account the number of kits needed for a (large) class, the workspace and expense may become limitations.

B. Algorithmic Robotics Curriculum

Robotics, in particular algorithmic robotics, is a rapidly developing field. Therefore, an algorithmic robotics class must teach students the fundamental concepts and necessary skills to understand and apply results from the state-of-the-art. A metric for such learning outcome would be students' ability to understand the digest of a major robotics conference (e.g., *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*). The curriculum structure would be based on the major science themes as identified in the area and technical committee divisions that are used as part of the editorial process at major conferences with care to separate the technical ("science") areas (e.g., control, perception, learning, and planning) and application ("system") domains (e.g., medical and life sciences, industrial robotics and automation, field robotics, etc.).

To achieve the above goal, learning should be focused on the ideas and underlying concepts behind the methods, rather than the methods per se. For example in motion planning, it is possible to use a bevy of methods to solve the problem of moving object - moving obstacle problem, including potential fields, rapidly exploring random tree (RRT) methods, etc. However, instead of only exposing students to such methods, we need to expose them to the underlying principles and ideas, such that students not only able to use these methods, but even more importantly, able to use the methods appropriately, selecting the right method for the right problem and understanding why the particular method is appropriate.

C. Learning Goals

CHARM aims to provide a robotic platform that allows undergraduate and postgraduate students explore different concepts in engineering. This robot not only tries to accommodate simple applications such as forward and inverse kinematics but also tries to provide students with a platform on which velocity control of robot arm is possible. This enables students to conduct motion planning in dynamic environment as opposed to environments with static targets Computer vision utilities of the robot allow students to practice perception, localization and velocity estimation of targets on workspace. This opens up more possible areas for students to explore.

The kinematic features of the robot arm along with computer vision utility allow motion planning on up to a 7-dimensional with moving targets on workspace. These strong features of CHARM let students use this robot to explore motion planning, vision and kinematics from elementary to advanced. Therefore the most important goal in this project is to provide students with a platform that does not involve the limitations posed by other kits. The learning by doing also allows for team projects allowing students from various disciplines to work together and develop team work qualities such as communication, collaboration and leadership.

V. CONCLUSION

The evolution of robotics and automation has resulted in the growing significance of training and educating the future engineers, researchers and technicians with a good grasp of mechatronic systems. Many of the topics in this area can be mastered through learning by doing. Robots are naturally one of the main tools in teaching robotics and mechatronics and they play an increasingly important role in education from high school to postgraduate studies. Their application does not stop at robotics education but also extends to other areas such as mathematics and physics. For this reason development of suitable robotic kits can prove to be very useful for students interested in robotics.

In summary, the CHARM robot platform has been developed as an example of a flexible, yet systematic, learning and research tool in the study of robotics concepts including robot arm kinematics, motion planning, and computer vision. The robot platform aims to alleviate the limitations in other kits by providing velocity control in robot arm, computer vision and testbed that allows for implementing motion planning concepts in time varying environments with a very good quality. Therefore we believe this robot platform will serve as a more suitable learning tool for undergraduate and postgraduate university students by allowing innovation and learning by doing what is appropriate for university level students.

In future work, the coin sorting problem can be extended to consider the fully optimal time optimal solution for the coin moving problem, or potentially the coin sorting problem under sensor or actuator uncertainty.

REFERENCES

[1] J. Weinberg and X. Yu, “Robotics in education: Low-cost platforms for teaching integrated systems,” *Robotics Automation Magazine, IEEE*, vol. 10, no. 2, pp. 4 – 6, June 2003.

[2] Z. Dodds, L. Greenwald, A. Howard, S. Tejada, and J. Weinberg, “Components, curriculum, and community: Robots and robotics in undergraduate ai education,” *AI magazine*, vol. 27, no. 1, p. 11, 2006.

[3] S. P. N. Singh, R. Fitch, and S. Williams, “A research-driven approach to undergraduate robotics education,” *Computers in Education Journal*, vol. 1, no. 4, pp. 21–27, 2010.

[4] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, “The e-puck, a robot designed for education in engineering,” in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, no. 1, 2009, pp. 59–65.

[5] M. W. Spong and D. J. Block, “The pendubot: A mechatronic system for control research and education,” in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 1. IEEE, 1995, pp. 555–556.

[6] W. E. Dixon, D. M. Dawson, B. T. Costic, and M. S. De Queiroz, “A matlab-based control systems laboratory experience for undergraduate students: toward standardization and shared resources,” *Education, IEEE Transactions on*, vol. 45, no. 3, pp. 218–226, 2002.

[7] R. Mitnik, M. Nussbaum, and A. Soto, “An autonomous educational mobile robot mediator,” *Autonomous Robots*, vol. 25, no. 4, pp. 367–382, 2008.

[8] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. C. Lovett, and M. K. Norman, *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons, 2010.

[9] S. Sheppard, K. Macatangay, A. Colby, and W. M. Sullivan, *Educating engineers: Designing for the future of the field*. Jossey-Bass San Francisco, CA, 2009, vol. 9.

[10] R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik, “Classification of coins using an eigenspace approach,” *Pattern Recognition Letters*, vol. 26, no. 1, pp. 61–75, 2005.

[11] M. Furst, G. Kronreif, C. Wogerer, M. Rubik, I. Hollander, and H. Penz, “Development of a mechatronic device for high-speed coin sorting,” in *Industrial Technology, 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 185–189.

[12] G. Wilfong, “Motion planning in the presence of movable obstacles,” *Annals of Mathematics and Artificial Intelligence*, vol. 3, no. 1, pp. 131–150, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF01530890

[13] T. M. Nguyen, S. Ahuja, and Q. J. Wu, “A real-time ellipse detection based on edge grouping,” in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE, 2009, pp. 3280–3286.

[14] S. Galvan, D. Botturi, A. Castellani, and P. Fiorini, “Innovative robotics teaching using lego sets,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 721–726.

[15] M. Rosenblatt and H. Choset, “Designing and implementing hands-on robotics labs,” *Intelligent Systems and Their Applications, IEEE*, vol. 15, no. 6, pp. 32–39, 2000.

[16] A. G. Alleyne, D. J. Block, S. P. Meyn, W. R. Perkins, and M. W. Spong, “An interdisciplinary, interdepartmental control systems laboratory,” *Control Systems, IEEE*, vol. 25, no. 1, pp. 50–55, 2005.

VI. APPENDIX 1: VIDEO ATTACHMENT

A video of the CHARM robot sorting coins has been submitted as part of this paper. It is also online at: `http://youtu.be/hH5_9t7lPhQ`.