International Journal of Medical, Medicine and Health Sciences
ISSN: 2517-9969
Vol:12, No:4, 2018

Mathematical Description of Functional Motion and Application as a Feeding Mode for General Purpose Assistive Robots

Martin Leroux, Sylvain Brisebois

Abstract**—**Eating a meal is among the Activities of Daily Living, but it takes a lot of time and effort for people with physical or functional limitations. Dedicated technologies are cumbersome and not portable, while general-purpose assistive robots such as wheelchair-based manipulators are too hard to control for elaborate continuous motion like eating. Eating with such devices has not previously been automated, since there existed no description of a feeding motion for uncontrolled environments. In this paper, we introduce a feeding mode for assistive manipulators, including a mathematical description of trajectories for motions that are difficult to perform manually such as gathering and scooping food at a defined/desired pace. We implement these trajectories in a sequence of movements for a semi-automated feeding mode which can be controlled with a very simple 3-button interface, allowing the user to have control over the feeding pace. Finally, we demonstrate the feeding mode with a JACO robotic arm and compare the eating speed, measured in bites per minute of three eating methods: a healthy person eating unaided, a person with upper limb limitations or disability using JACO with manual control, and a person with limitations using JACO with the feeding mode. We found that the feeding mode allows eating about 5 bites per minute, which should be sufficient to eat a meal under 30min.

Keywords**—**Assistive robotics, Automated feeding, Elderly care, Trajectory design, Human-Robot Interaction.

I. Introduction

EATING a meal is among the Activities of Daily Living (ADL) [1], but it is a difficult task for people with mobility and functional limitations, such as cerebral palsy or various levels of paralysis. For a meal to be enjoyable, it must be eaten at a reasonable pace with minimal effort [2]. Dedicated technologies, such as exoskeletons [3] and feeding robots, have been developed to assist these patients in eating independently [4]–[6], thus reducing the burden on caregivers.

Although many automated feeding devices are performing well [6]–[9], their operation relies on an almost perfectly controlled environment. This usually means that food is placed in a special plate, often attached to the robot to fix its location, and is picked up using a dedicated utensil. This makes it easier to hard-code trajectories that will pick up food every time, but makes self-feeding impossible as soon as any part of the environment is modified. These solutions are over-specialized (task-specific and tools-specific), often cumbersome and limiting, notably in terms of portability. These factors usually prohibit their use outside of the users home.

General-purpose assistive robots, such as wheelchair-based manipulators, are another option [10], [11]. They can be used for performing a variety of tasks [12], but these robots are hard to control for elaborate continuous motions such as eating [13]. Sustained effort and focus are required to eat a meal by manually controlling such a device; little to no automation is available for any given task. However, these robots are not limited to heavily controlled environments, making them significantly more versatile. Despite this potential, to our knowledge, no formal description of a scooping motion (nor other functional motions) was ever developed for free, uncontrolled environments.

The aim of this paper is to develop a semi-automated feeding mode for general purpose assistive manipulators which allows a user to eat a meal at a reasonable pace and with limited effort. This mode should offer enough flexibility to function for multiple plate locations and sizes. In the following sections, we will first present a mathematical framework for designing trajectories that will allow for this flexibility, as well as parameters mathematically defining scooping and food gathering motions to use as trajectories for manipulators. Then we will propose a sequence of motions for semi-automated feeding and integrate it with a simple 3-button interface. Finally, we will demonstrate our mode using a JACO robot (Kinova Robotics, Canada) to compare the pace of eating (in bites per minutes) while using the feeding mode to that of a healthy person and of a disabled person manually controlling the robot.

II. Configuration Independent Motion

In order to allow functional motions such as scooping to be adaptable to the plate location, the motion must be described independently of the robot configuration. Usually, manipulator trajectories are designed either as joint positions or as Cartesian coordinates relative to an immobile reference frame, which is dependent on the robot configuration and, therefore, usual trajectory descriptions cannot be used for functional motions. In this section, we present a framework to create trajectories expressed in the tool frame, which are consequently independent of the robot configuration as long as the motion does not cross problem-inducing points like kinematic singularities or self-collisions.

A. Tool-Based Trajectories

Since robots are usually designed to receive inputs (either positions or velocities) in the base frame, the most simple way

Martin Leroux and Sylvain Brisebois are with Kinova Robotics, Canada (e-mail: mleroux@kinova.ca).

to send tool-based trajectories is to express them in terms of velocities in the tool frame of reference and then to project said velocity in the base frame. Many robots would also accept inputs in the joint space (positions, velocities, torques). All joints must be controlled, so it is clear that any trajectory designed in the joint space would be configuration-dependent, which we want to avoid.

1) End-Point Control: The position of the tool frame relative to the inertial frame can be expressed in homogeneous coordinates through a transformation matrix given by (1).

$$\boldsymbol{T}_{4\text{x}4} = \begin{bmatrix} \boldsymbol{R}_{3\text{x}3} & \boldsymbol{p}_{3\text{x}1} \\ \boldsymbol{0}_{1\text{x}3} & 1 \end{bmatrix}, \qquad (1)$$

where:

- $\boldsymbol{R}$ is the rotation matrix representing the orientation of the tool relative to the inertial frame.
- $\boldsymbol{p}$ is the position of the tool relative to the inertial frame.

Then, a velocity along an arbitrary axis $\boldsymbol{v_u}$ expressed in the tool reference frame is given in the inertial frame by (2).

$${}^{\boldsymbol{b}}\boldsymbol{v_u} = \boldsymbol{R}\boldsymbol{v_u} \qquad (2)$$

All that remains is to express $\boldsymbol{v_u}$ as a function of time to describe a desired trajectory. However, it can be hard to visualize trajectories in terms of velocities, so we propose a method for designing trajectories with positions and then taking the time derivative to obtain the desired velocities.

2) Planar Trajectories: Functional trajectories such as scooping can usually be described as a constant forward velocity in the tool frame with a varying orientation, which simplifies the trajectory description to a function of orientation over time and a fixed value of $\boldsymbol{v_u}$ in (2).

For a trajectory described by a polynomial, for example in the ZX plane of the tool frame, such as in (3)

$$x(z) = \Sigma_{i=0}^{n} a_i z^i, \qquad (3)$$

the orientation about the normal axis (here Y) is given by (4):

$$\theta = \arctan(\frac{dx}{dz}) = \arctan(\Sigma_{i=1}^{n} a_i i z^{i-1}) \qquad (4)$$

By taking the time derivative, we obtain the angular velocity.

$$\omega = \frac{\Sigma_{i=2}^{n} a_i i(i-1) z^{i-2}}{1 + (\Sigma_{i=1}^{n} a_i i z^{i-1})^2 \dot{z}} \qquad (5)$$

To generalize this method to a 3D trajectory, simply describe two perpendicular planar trajectories (say in the ZX and ZY planes) and add the angular velocities, which are orthogonal to each other.

Finally, the forward velocity for performing the motion in a given time t can be set taking the curve length as in (6):

$$L = \int_{0}^{z_{max}} \sqrt{1 + (\frac{dx}{dz} + \frac{dy}{dz})^2} dz \qquad (6)$$

Then, the required forward velocity is simply given by (7).

$$v = \frac{L}{t}, \qquad (7)$$

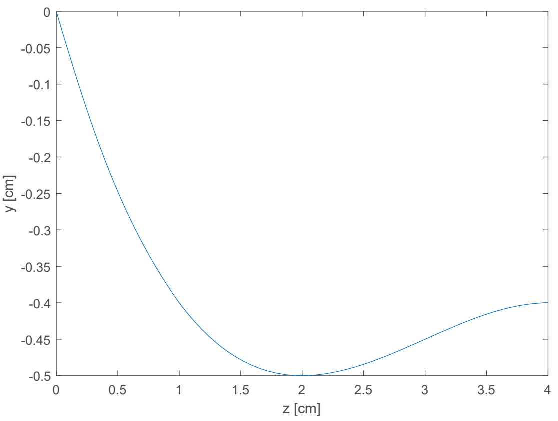

Fig. 1 3rd order polynomial for scooping trajectory

and $\dot{z}$ is given by (8).

$$\dot{z} = v \cos(\theta) \qquad (8)$$

B. Scooping

The scooping motion can be defined using the method described above. Assuming the scooping motion is in the ZY plane with the positive Z-axis oriented ”forward”, we are looking for a polynomial $y(z)$ with the following characteristics:

- $y(0) = 0$
- $y(2cm) = -0.5cm$
- $y(2cm)$ is a minima
- $\frac{dy}{dz}|_0 < 0$

This set of conditions was chosen empirically to describe a motion going down $0.5cm$ while going forward $2cm$ and rotating until the utensil is horizontal, based on our observations of a normal-looking scooping motion. These characteristics are respected by (9), which is plotted in Fig. 1.

$$y(z) = -0.025z^3 + 0.225z^2 - 0.6z \qquad (9)$$

Although it would seem appropriate to make use of a remote center of rotation to make certain that the trajectory is followed by the tip of the utensil, this would add constraints to the system, such as forcing a certain length of tool or requiring it to be measured and input in the system every time. Since the distance between the tip of the tool and the center of rotation is normally short, and the motion is planar in the gripper frame, not using the remote center of rotation does not distort the motion to a significant degree. Therefore, to take advantage of the added flexibility of the system, we recommend not using a remote center of rotation.

C. Food Regrouping

After eating for a while, the remaining food will be scattered more or less randomly around the plate, further and further reducing the odds that scooping at a given location in the plate will grab a satisfying amount of food. As its name suggests, the purpose of the food regrouping motion is to gather the remaining food and bring it back to the center of the plate, where it can be more easily scooped. To achieve this, the main required motion is to follow the inner edge of the plate with a utensil perpendicular to the plate edge and describing a circular motion.

Although such a motion could be expressed in terms of polynomial derivatives, a perfectly circular motion can be obtained with simpler steps.

Given a motion speed $v[cm/s]$, following the edge of half a plate of radius $R[cm]$ will take:

$$t = \frac{\pi R}{v}[s] \tag{10}$$

During this time, the tool orientation must rotate of π, yielding a necessary angular velocity of:

$$\omega = \frac{\pi}{t}[rad/s] \tag{11}$$

III. Feeding Sequence

The general idea for the feeding mode is to mimic the eating behaviour of a healthy person by scooping at various places in the plate, regrouping the food in the center of the plate when it becomes sparse, and then resuming scooping until the plate is mostly empty. In this section, we present the initialization, eating and regrouping sequences designed to make an intuitive and efficient robot behaviour.

A. Preparing the Motion

In order for the various motions to be automated correctly, some information must first be sent to the robot.

1) *The position of the user*
The location of the user's head must be known to the robot in order for the food to be brought at a convenient place for eating. Since most users of assistive robotics are in a wheelchair, the location of their head is unlikely to move significantly between each use of the feeding mode. Therefore, this location could be recorded once as a custom parameter for the user on the first use and then remembered for all future uses of the feeding mode. Alternatively, the position could be adjusted by manually positioning the robot when launching the mode. This would allow for some additional portability if the robot can be detached from its base.

2) *The plate position*
One of the objectives of our feeding mode is to be flexible regarding the location of the food. This means not only that there must be some tolerance between its estimated location and its actual location, but also that the plate may be located almost anywhere in the workspace of the robot, rather than precisely fixed to it as it is with other single-purpose feeding robots [8]. The plate position may be recorded by manually placing the robot at the appropriate position before eating. If a vision system is available and able to identify the plate, this position could be acquired automatically [14], [15], thus reducing the burden on the users.

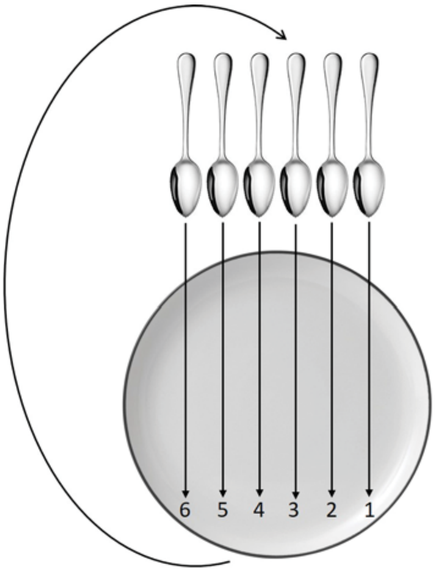

Fig. 2 Plate location offset for easier food scooping

3) *The plate inner radius*
As seen in (10), the inner radius of the plate is required to plan the correct food regrouping motion. If a dedicated standardized plate is associated with the product, its radius could be hard-coded in the program for the feeding mode. Otherwise, two methods may be used to estimate the plate radius. If a vision module is available, the radius may be readily measured with circle fitting for 2D vision or with direct measurements for 3D vision. If not, the user could manually record three or more locations on the inner plate edge and circle fitting may be used to estimate both the location of the plate center and its radius.

Assuming the assistive robot responsible for the motion has a positioning accuracy of at least $1cm$, any normal kind of plate or utensil can be used with our feeding mode. However, we find that the scooping motion is more easily performed with a spoon and that the food gathering motion is most efficient if the utensil used is somewhat flexible and if the plate borders are as high as the head of the utensil.

B. Eating Sequence

The eating sequence consists of three consecutive movements: going to the plate, scooping, and going to the mouth of the user. This results in 3-steps cycles that are intuitively predictable and simplify the interface.

The first step simply involves bringing the utensil to the plate. However, if the utensil were to go to the same location each time, there would quickly not be any more food to be scooped. To avoid this problem, every cycle moves the target location sideways relative to the utensil. This way, we take advantage of the fact that the scooping motion may push aside some food in the following cycle. This motion is illustrated in Fig. 2.

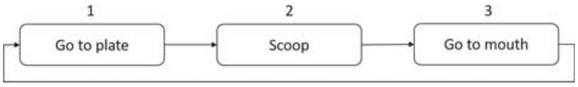

Fig. 3 Eating cycle

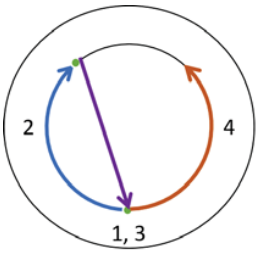

Fig. 4 Steps of the food regrouping sequence

Once positioned at the right location in the plate, the second step, consisting of the scooping motion described in the previous section, is performed. In order to avoid dropping food, the motion is interrupted once the utensil is horizontal, and then the robot lifts the utensil without changing orientation. Once the motion is complete, the third step is to bring the food to the previously recorded location of the user's head. Finally, the cycle goes back to step one. Each step is executed one at a time upon an input from the user (See Section IV).

The steps of the cycle are illustrated in Fig. 3.

C. Regrouping Sequence

The objective of the regrouping sequence is to gather the food on the edges of a plate and center it for an easier access during the scooping motion. This is performed in 4 steps, as illustrated in Fig. 4.

1) Go to the bottom of the plate and orient the utensil perpendicular to the radius of curvature of the plate, facing left.
2) Perform the food regrouping motion described in the previous section.
3) Go back to the bottom of the plate and orient the utensil facing right.
4) Perform the food regrouping motion in the opposite direction.

IV. User Interface

Since the users of assistive technologies have functional limitations, it is important that the interface of the feeding mode requires minimal effort as well as little to no dexterity. However, it is also important that the user still feels in control, so a fully automated motion is not an acceptable solution. A user would not want the pace of eating to be dictated by the robot, as this would likely feel extremely invasive. For this reason, we created a user interface that requires only three buttons to be pressed, as illustrated in Fig. 5. In order to make the user feel in complete control and safe, the robot can only move while a button is pressed.

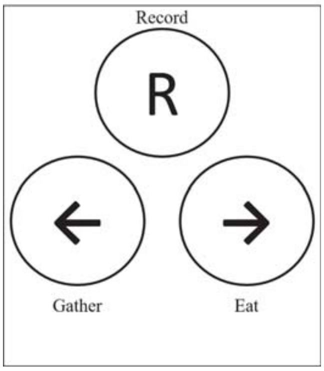

Fig. 5 3-button interface to the feeding mode

The purpose of the *Record* button is only to register points used in the initialization step. To identify which point is being recorded, the configuration of the robot may be used. For example, since the user's head should always be significantly higher than the plate, a threshold value on the vertical position of the end-effector of the robot is used to recognize the point as the position of the mouth. Also, if the utensil is vertical, as in ready-to-scoop, the position is recognized as the plate location.

Every press of the *Eat* button performs one step of the eating cycle; the actual action of eating requires only a single button to be pressed repeatedly.

Finally, the *Gather* button can first be used as a ”Go back” function, always interrupting the eating cycle to go back to the plate. This can be useful in case the scooping motion fails to grab food in the plate, this way there is no need to complete the cycle before trying another scoop. Upon two consecutive presses, the robot prepares for the food gathering motion by going to the bottom of the plate. Finally, a third consecutive press of the button launches the food regrouping sequence. The preparation and food gathering sequence were divided by a button press to allow the user to assert that the utensil is positioned correctly, but motions could be executed consecutively with the single press of the button.

V. Implementation and Results on a JACO Assistive Manipulator

JACO is a robotic manipulator developed by Kinova Robotics [16] designed to be used as an assistive device. It is a serial 6 degrees-of-freedom (DoF) robot with a seventh DoF to open and close under-actuated fingers. Since these fingers are oversized for fine manipulation, such as holding utensils, the robot also has an easier to grab tool holder, as seen in Fig. 6.

Even if, by using this tool, JACO can be manually controlled to eat autonomously, most users do not because it requires an unreasonable amount of effort and time. Some users even reported it taking over an hour [17] due to the difficulty of manually operating the robot for such fine movements. This is the main reason for the present work: we want users to actually be able to eat with their robots.

The implementation of our feeding motion for JACO was constrained by its wrist design. JACO's wrist is composed

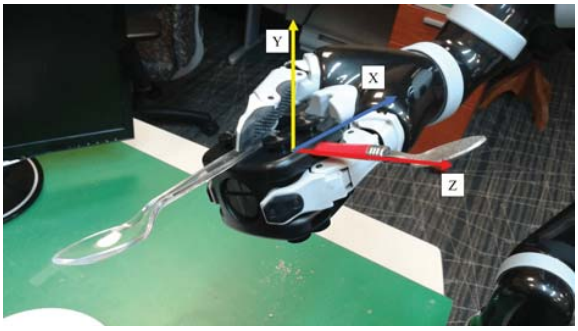

Fig. 6 JACO's dedicated small tool holder

of two 60° joint, designed to avoid the possibility of closing the wrist on itself, contrary to most 6+-DoF robots, which usually have a spherical wrist. This design makes the dexterous workspace much less symmetrical and harder to navigate. The main consequence of this is that pure rotation of the tool coordinate system is very limited in all directions but the roll-axis. Since the scooping and regrouping motions require rotations about different axes, it was deemed easier to use two mutually orthogonal utensils at the same time (as seen in Fig. 6), one for scooping and the other for gathering.

VI. Evaluation

As a metric to evaluate our feeding mode, we compare the maximum number of scoops/bites per minute a user can take. The maximum number of bites per minute was deemed a good metric to compare feeding strategies because it encompasses both the scoop success rate as well as the motion velocity and efficiency.

As a benchmark, we measured the pace of eating of two healthy members of our research group. We also report the pace of eating using the manual control (via joystick) of JACO as estimated by a user with upper limb physical limitations. Finally, we measured the pace of eating using our 3-buttons interface for the feeding mode. Each person was served 1.5 cup of oatmeal. The pace was acquired over the time required by each individual to finish eating the content of the plate. Oatmeal was used because it has a similar texture as other clumpy meals that are easier to eat for people with functional limitations.

The maximum number of bites per minute was deemed a good metric to compare feeding strategies because it encompasses both the scoop success rate as well as the motion velocity and efficiency. Moreover, it is an indication of the total time required to eat a meal, which is considered to be the main issue with the use of general purpose assistive devices.

Total time to eat on the other hand would not have been appropriate because it would depend on many uncontrollable factors such as the user's appetite or habits while eating. A hungry person may eat very fast while another person may take a lot of time because he or she is having a discussion at the same time, which would yield high variability in the data.

Our comparative results are summarized in Table I.

TABLE I
Maximum Number of Bites Taken per Minute

Healthy person	JACO manual control	Feeding mode (excluding Regrouping)	Feeding mode (including regrouping)
5-10 min^{-1}	0-4 min^{-1}	5-7 min^{-1}	3-5 min^{-1}

VII. Discussion

We aimed to create a feeding mode for general-purpose assistive robots to allow users to eat a meal at a reasonable pace, with minimal effort and which offered enough flexibility to be used outside of the user's often heavily controlled home. In this section, we discuss the advantages and limitations of our motion planning and user interface as well as the results obtained from our implementation on JACO to assess our success.

It is worth noting the value of having this kind of mode on a general purpose assistive robot rather than a specialized one. Assistive technologies can be very expensive, so having a single robot for multiple tasks costs less than having a specialized machine for every single task. Moreover, one cannot expect users to carry all their specialized tools everywhere. General-purpose robots are simply more reliable in uncontrolled environments.

The main contribution of this paper is the functional motion mathematical framework and the mathematical description of the scooping motion. This knowledge may be used in more assistive applications, for example more fully automated feeding modes, or to perform other similar tasks. The configuration-independent trajectory framework could be used to perform other useful functional tasks such as opening doors or shaving. Our results regarding the pace of eating are not statistically significant given our small number of samples, but are only meant to be a demonstration of the usefulness of the scooping motion.

A. Motion Planning

The scooping motion we propose can be performed from any direction and in any starting position in the workspace of the robot, thus offering unprecedented flexibility. This means that users could go to the restaurant, have their plate placed anywhere usual, find a configuration for their manipulator which reaches the plate with minimum nuisance to himself and his seating neighbours and proceed. This could not be achievable with a single-purpose assistive device which relies on a controlled environment.

On the other hand, the proposed trajectory heading toward the tip of the utensil, although functional, requires a lot of space around the robot to be performed. A sideways scooping motion could be performed within a smaller volume, but was deemed more complex to implement correctly due to the direction of the rotation and to the less easily visualized nature of the required motion.

The food regrouping motion was deemed adequate during our tests. By including this motion, the vast majority of a meal may be eaten without leaving a significant amount in the plate. However, larger plates would require a more elaborate motion, because the width of the utensil becomes negligible relative to the radius of the plate. For example, successive food gathering trajectories could be performed with gradually decreasing trajectory radius.

B. *User Interface*

Our 3-button user interface is very simple to use and easy to understand. On one hand, pressing a button requires little effort compared to manually controlling a robot, which is often done through the use of a joystick or other inputs with less DoF than the robot [2]. This means that the user is not required to apply his entire focus on the task of eating and may at the same time interact with other people, which makes eating a meal significantly more enjoyable. In addition, the limited number of buttons makes it ideal for people with important physical limitations that prohibit the use of more elaborate controllers, since a button requires no dexterity to be pressed and may be located anywhere in the reach of the user. Moreover, the chosen sequences associated with the buttons makes the use of a ”mode-switching” button, often used when the number of input sources is limited, unnecessary, thus making the system much simpler for that customer base. This is important because mode switching is considered a major hurdle for the use of semi-automated assistive technologies [18].

The main limitation of our interface is the necessity of a recording action when initializing the feeding mode. Although it only has to be performed once, the manual control of the robot is deemed tedious during the initialization because most of it cannot be done ahead of time. This means that, at a dinner table, when everyone is served, the user still has to take what could be a significant amount of time before he is able to start eating. As mentioned earlier, this could be further automated with the use of a vision module which could see and compute all the required locations. Given the advancements in artificial intelligence, image segmentation and image processing in recent years [19], we can assume that this will not remain a challenge as soon as dedicated hardware is included in the process. Moreover, vision would have many more uses as an upgrade to wheelchair-mounted arms and is clearly the next major progress in assistive robotics [20].

It has also come to our attention that not all users of assistive robots can hold a button for a continuous time, for example because of uncontrollable spasms. With that in mind, it would be necessary to make the button holding feature, which was implemented to empower the user with a greater feeling of control, optional. Some users may also require inputs to be adapted to their capabilities to have fully access to these new functionalities.

C. *Eating Pace Evaluation*

When comparing the maximum number of bites per minute using the feeding mode to that estimated for a healthy person, we can see that the feeding mode is still only about half the speed. This is due to two main reasons. First, assistive robots are inherently slow (they are usually limited to 15 cm/s) to make them easier to control, but most of all due to safety reasons. This means that even assuming a scooping success rate of 100%, the robot arm could not reach the full speed of a healthy human arm. This is specially true during the food regrouping motion which is performed slowly and all around the plate while a person could very well limit the gathering to a smaller space for better efficiency. With other safety features, such as fast collision feedback, the speed of the robot could be significantly increased. Second, a healthy person has a significant advantage over an assistive robot because he or she has two arms which allow for more efficient motion, for example to push food over a fork with a knife when scooping is hard. It also allows for multi-tasking, such as eating with one hand and gathering food in the middle of the plate with the other. However, adding another assistive arm to take advantage of these benefits is not realistic for monetary reasons, but also because the system would become very cumbersome and would lose some of its portability. Even including the food gathering motion, the pace of eating using our interface is akin to that of a healthy person having a sustained conversation while eating, which is possible with the feeding mode due to our simple interface.

Additionally, it is worth noting that the eating speed is significantly higher using the semi-automated feeding mode rather than manual control. The time required to eat a meal no longer feels prohibitive. Given this pace, one should be able to eat a meal under 30 minutes, as a healthy person eating slowly would.

VIII. Conclusion

In this paper, we proposed a feeding mode for general-purpose assistive robots that can be used in an uncontrolled environment, with almost any plate size and location, by defining useful motions as velocities in the tool frame of the robot. The mode can be used even for people with important motor limitations through our 3-button interface, which allows feeding to be performed as a sequence without having to keep focus on the task of eating. Our implementation of the feeding mode on a JACO robot showed that this mode allowed eating a meal at a similar pace to that of a healthy person. The integration of a vision system, enabling the the robot to capture some information about its environment, could allow for an even smoother experience by removing the need for manually recording positions and dimensions, which takes time and effort for a typical user of assistive technology. This would also allow the user interface to be simplified to two buttons.

Our main contribution is the elaboration of a mathematical framework for designing functional trajectories. With the mathematical description of motions such as scooping, as presented here, researchers will be able to more readily provide useful functionalities for assistive robots. The results of such work will empower users with more autonomy in their daily lives.

Acknowledgment

The authors would like to thank Sebastien Lepage for his feedback and help as a regular user of assistive technology.

REFERENCES

[1] D. Foti and J. S. Koketsu, “Activities of daily living,” *Pedrettis Occupational Therapy: Practical Skills for Physical Dysfunction*, vol. 7, pp. 157–232, 2013.

[2] I. Naotunna, C. J. Perera, C. Sandaruwan, R. Gopura, and T. D. Lalitharatne, “Meal assistance robots: A review on current status, challenges and future directions,” in *System Integration (SII), 2015 IEEE/SICE International Symposium on*. IEEE, 2015, pp. 211–216.

[3] W. T. Latt, T. P. Luu, C. Kuah, and A. W. Tech, “Towards an upper-limb exoskeleton system for assistance in activities of daily living (adls),” in *Proceedings of the international Convention on Rehabilitation Engineering & Assistive Technology*. Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre, 2014, p. 12.

[4] S. Ishii, S. Tanaka, and F. Hiramatsu, “Meal assistance robot for severely handicapped people,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2. IEEE, 1995, pp. 1308–1313.

[5] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, “The role of assistive robotics in the lives of persons with disability,” *American Journal of Physical Medicine & Rehabilitation*, vol. 89, no. 6, pp. 509–521, 2010.

[6] W.-K. Song and J. Kim, “Novel assistive robot for self-feeding,” in *Robotic Systems-Applications, Control and Programming*. InTech, 2012.

[7] J. J. Villarreal and S. Ljungblad, “Experience centred design for a robotic eating aid,” in *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*. IEEE, 2011, pp. 155–156.

[8] OBI. (2016) Tech specs. (Online). Available: https://meetobi.com/tech-specs/.

[9] M. J. Topping and J. K. Smith, “The development of handy 1. a robotic system to assist the severely disabled,” *Technology and Disability*, vol. 10, no. 2, pp. 95–105, 1999.

[10] M. Topping, “Flexibot–a multi-functional general purpose service robot,” *Industrial Robot: An International Journal*, vol. 28, no. 5, pp. 395–401, 2001.

[11] H. Kwee and C. Stanger, “The manus robot arm,” *Rehabilitation Robotics Newsletter*, vol. 5, no. 2, pp. 1–2, 1993.

[12] F. Routhier, P. Archambault, M. Cyr, V. Maheu, M. Lemay, and I. Gélinas, “Benefits of jaco robotic arm on independent living and social participation: an exploratory study,” in *RESNA Annual Conference*, 2014.

[13] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier, “Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities,” in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[14] Y. Cheng, X. Zhao, R. Cai, Z. Li, K. Huang, and Y. Rui, “Semi-supervised multimodal deep learning for rgb-d object recognition.” in *IJCAI*, 2016, pp. 3345–3351.

[15] J. Lahoud and B. Ghanem, “2d-driven 3d object detection in rgb-d images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4622–4630.

[16] A. Campeau-Lecours, V. Maheu, S. Lepage, H. Lamontagne, S. Latour, L. Paquet, and N. Hardie, “Jaco assistive robotic device: Empowering people with disabilities through innovative algorithms,” in *Rehabilitation Engineering and Assistive Technology Society of North America (RESNA) an-255 nual conference*, vol. 14, 2016.

[17] C.-S. Chung, H. Wang, and R. A. Cooper, “Functional assessment and performance evaluation for assistive robotic manipulators: Literature review,” *The journal of spinal cord medicine*, vol. 36, no. 4, pp. 273–289, 2013.

[18] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, “Assistive teleoperation of robot arms via automatic time-optimal mode switching,” in *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*. IEEE, 2016, pp. 35–42.

[19] X. Xie, M. Jones, and G. Tam, “Recognition, tracking, and optimisation,” *International Journal of Computer Vision*, vol. 122, no. 3, pp. 409–410, 2017.

[20] H. Jiang, J. P. Wachs, and B. S. Duerstock, “Integrated vision-based robotic arm interface for operators with upper limb mobility impairments,” in *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.