Title
Modeling of Joint Synergy and Spasticity in Stroke Patients to Solve Arm Reach Tasks

Authors
Feldman, Aaron
Shen, Yang
Rosen, Jacob

Publication Date
2017-12-02

Data Availability
The data associated with this publication are available upon request.

Peer reviewed

Modeling of Joint Synergy and Spasticity in Stroke Patients to Solve Arm Reach Tasks

Aaron Feldman
Milken Community School
Los Angeles, United States
afeldman642@milkenschool.org

Yang Shen
Dept. of Mechanical and Aerospace Eng.
University of California at Los Angeles
Los Angeles, United States
yangshen@ucla.edu

Jacob Rosen
Dept. of Mechanical and Aerospace Eng.
University of California at Los Angeles
Los Angeles, United States
jacobrosen@ucla.edu

***Abstract*—Stroke rehabilitation often focuses on counteracting stroke-induced muscle weakness so as to allow patients to regain the ability to complete activities of daily living. However, other motor issues such as deleterious joint synergies (involuntary coactivation of joints) and spasticity (involuntary muscular contractions that limit range of motion) may impede stroke victims and thus necessitate modification of the training methods utilized in rehabilitation. To increase the rehabilitation efficacy and help patients relearn how to complete essential reach tasks, alternative routes that account for the synergy and spasticity must be identified. The procedure for finding efficient reach task paths is accomplished by performing a branched iterative search. Each path is scored based on the distance from the final hand position to the reach task target as well as the total travel distance of the path. Synergy is included in the algorithm by multiplying the joint command, the desired joint change, by a matrix of synergy interactions. Spasticity is included by either imposing hard joint angle limits or by specifying a spastic region through which movement induces score penalties. The algorithm successfully reduces travel through spastic regions while counteracting synergistic deviations that would increase path travel distance.**

Keywords—stroke rehabilitation, synergy, spasticity, arm model, reach task, path solving algorithm

I. Introduction

Stroke affects approximately 795,000 individuals per year in the United States [1]. The neurological damage sustained to these individuals can result in motor control impairment [2]. Among the resulting issues experienced are joint synergy and spasticity. While joint synergy, involuntary coactivation of joints, is also present in healthy individuals, neurological damage can cause deleterious synergies to develop. These deleterious synergies may interfere with motion and prevent certain arm configurations [3], [4]. Spasticity refers to persistent and involuntary muscular contraction that results in stiffness and hampers movement of the limb. The percentage of stroke victims experiencing a classifiable degree of spasticity ranges from 19-42% [5], [6]. Physical rehabilitation can treat stroke-induced muscle weakness, synergy, and spasticity [7]-[9]. Task-specific therapies that simulate activities of daily living (ADL) can be used to relearn and regain the requisite motions [10]. However, individuals with joint synergy and spasticity may not know how to complete daily living tasks while compensating for their motor issues. Thus, unable to complete ADLs that could be possible for them, they would have suboptimal rehabilition. This research focuses on developing a needed algorithm that finds efficient reach paths by taking into account the joint synergy and spasticity. The algorithm minimizes travel through spastic regions so as to avoid pain and also counteracts synergy to enable completion of the reach task. In addition to helping patients complete ADLs, the algorithm could serve as a tool for classifying the feasibility, difficulty, and painfulness of reach tasks as it tracks the joint degree change in the spastic region and assigns a score to each task solution. This classification could better inform therapists about the appropriateness of possible training tasks and help patients better understand their limitations and potential capabilities. Effective and entertaining therapies have been combined in virtual reality games that simulate ADLs [11], [12]; this algorithm could allow for the development of virtual reality games whose difficulty is tailored for the individual. To aid in rehabilitation, exoskeletons have been developed to provide force assistance and directional guidance for those who can generate arm movements but would otherwise be unable to complete a given reach task [13], [14]. Yet, those suffering from synergy and spasticity may not know how to effectively achieve a task, preventing them from appropriately guiding the exoskeleton. The algorithm could show the patient how to guide the exoskeleton and complete the task while avoiding pain. In the case of a patient suffering from muscular weakness, it could program the exoskeleton to follow the appropriate path.

II. Approach

A. Seven Degree of Freedom Denavit-Hartenberg Model

To simulate the reach tasks, a robotic representation of the arm was developed in MATLAB using the Robotic Vision toolbox [15]. This arm model, illustrated in Fig. 1, has seven degrees of freedom (DOF), with each joint corresponding to a different axis of rotation for the arm. There are three shoulder joints: shoulder interior rotation, shoulder abduction, and shoulder flexion, two elbow joints: elbow flexion and wrist/forearm pronation, and two wrist joints: wrist flexion and

ulnar deviation. This seven DOF model matches related work such as [16] that uses Denavit-Hartenberg (D-H) parameters. However, the D-H parameters are defined in TABLE I so that the arm points down when all joint angles are 0º, consistent with the Fugl-Meyer (FM) joint angle data collected in [4]. The length of the of the upper arm, lu, forearm, lf, and hand, lh, are 34, 27, and 20 cm respectively. Because these data inform the synergy model, the consistency between the data and the model was validated by passing the data through the model to recreate each task. This model of the seven DOF arm is also consistent with the exoskeleton EXO-UL7 described in [13], so that the arm movements can be encoded for the exoskeleton.

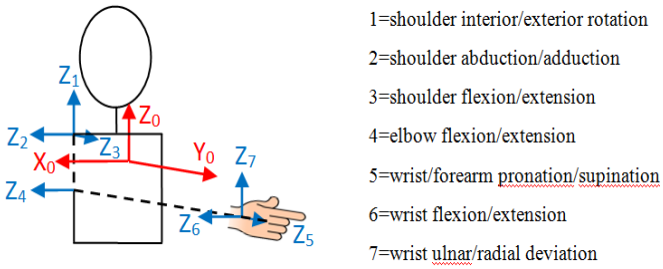

Fig. 1. Joint axes

TABLE I. D-H PARAMETERS

Link	θ Offset	D	A	α	Joint	Description
					1	Shoulder interior rotation
1	0	0	0	-π/2	2	Shoulder abduction
2	-π/2	0	0	-π/2	3	Shoulder Flexion
3	0	0	-lu	0	4	Elbow Flexion
4	-π/2	0	0	-π/2	5	Wrist/Forearm Pronation
5	0	-lf	0	π/2	6	Wrist Flexion
6	-π/2	0	0	-π/2	7	Wrist Ulnar Deviation
7	0	0	lh	0		Hand (Tool Tip)

B. *Algorithm Implementation*

The algorithm performs an iterative branched search in which nodes, representing joint configurations, are created, scored, and evaluated. The algorithm flow is diagrammed in Fig. 2. Initially, a single node stores the starting configuration of the seven joints. Given a node, an angular increment or decrement (set to 3º) is considered for each of its joints. These joint commands can cause up to fourteen "children" nodes to be created. The node data structure stores the current values of the seven joints (the node's joint vector), the sequence of joint commands that led to these joint values, and the current XYZ position of the hand. For example, the sequence [-4, 1, 2] means that, starting from the initial configuration, first joint 4 was decremented, then joint 3 was incremented, and lastly joint 2 was incremented. Based on the information stored in the node, the algorithm computes the travel distance for the XYZ path corresponding to the sequence of joint commands. Because the children nodes include all the information of the "parent" node, the latter is deleted once the children have been created.

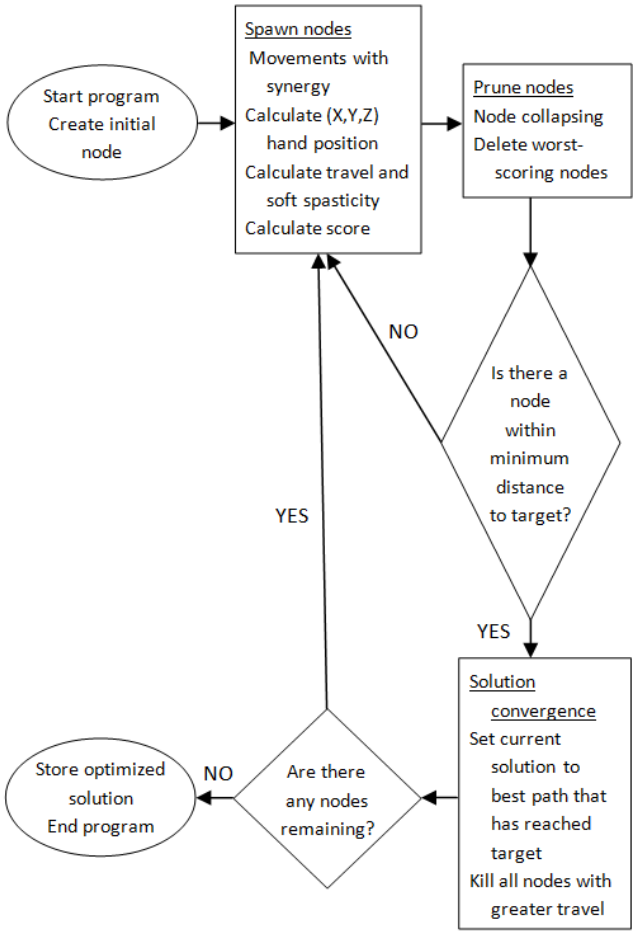

Fig. 2. Algorithm flowchart

C. *Synergy*

Each of the seven FM tasks for which data was collected in [4] primarily utilized just one of the seven joints. The data were analyzed in [4] to develop a synergy matrix whose accuracy was confirmed via a statistical comparison of synergy matrix and FM scores. To derive the synergy matrix, the least squares linear fit was calculated for every pair (i,j) of joint angles using their data over the entire set of tasks; the (i,j)th element of the synergy matrix was the slope of the corresponding linear fit. In this approach, the synergy matrix has values of one along the diagonal. To model the effects of synergy while constructing paths, our algorithm uses this synergy matrix. Given a command for joint i, the angular

change for all seven joints is the ith column of the synergy matrix scaled by the joint angular increment/decrement. To generate a more robust synergy matrix, the least squares fit was modified from that proposed in [4] to use all the data from each task (i.e., four trials instead of one).

D. Node Pruning and Scoring

Because any parent node can spawn up to fourteen children nodes, it was deemed too computationally intensive to track all possible children nodes as the program progresses. A scoring metric ensures that only the least successful paths are deleted. Once the number of nodes exceeds a fixed limit, set at 1000 in this work, the lowest-scoring paths are deleted. A node's score is a weighted sum of the distance to the target from the current hand position and the net path travel distance in XYZ space. Under the assumption that an optimal path minimizes travel distance while approaching the target, lower scores are better. The weighting of 2:1 for distance to target versus net travel distance favors task completion by favoring paths that approach the target. If this weight ratio does not result in the algorithm finding a solution, i.e., a path that reaches the target, the ratio is gradually increased until task completion is achieved.

E. Node Direction Rule

Given the linear synergy model, angular commutativity applies: reordering a sequence of commands does not change the final hand position. Consider a sequence of joint commands that includes both positive and negative increments for the same joint. Compared to the same sequence but with the opposing commands removed, the longer sequence will have the same final hand position but is likely to entail a greater travel distance. Accordingly, the algorithm limits node creation by not creating a new node if its last command would oppose any previous command.

F. Node Collapsing

The angular commutativity concept implies that if two nodes have similar joint vectors, then one is redundant. The less efficient node, the one with the greater travel distance, should be pruned. To achieve this pruning while remaining computationally efficient, node hand positions are quantized in XYZ space into eight cubic cm "boxes". Because nodes with similar joint vectors must have similar XYZ hand positions, only nodes within the same box need to be compared. If the joint vectors are similar, with no joint differing by more than 1º, the less efficient node is deleted.

G. Solution Convergence

When a solution is first found, indicated by the final hand position being within a minimum distance (5 cm) from the target, all nodes with longer travel distances are deleted as being less successful. Nodes with shorter travel distances have the potential to be more successful and therefore continue to spawn children. A descendant of these nodes will either reach the target with less travel distance, causing it to replace the previous solution, or will be deleted for exceeding the current solution's travel distance. This process converges to a single solution.

H. Spasticity

Two types of spasticity were added to the model to allow for flexibility in future applications. Hard spasticity defines joint limits that cannot be exceeded. All subjects have this type of limit. In the model, healthy human limits are based on the findings in [17]. For a patient suffering from spasticity, these limits could be modified based on the individual's range of motion to ensure that the solution will not exceed the individual's limits. Modeling these individual limits would enable classification of reach tasks as feasible or impossible, allow appropriate path selection when using an exoskeleton, and avoid excessive pain to the patient.

The model also includes a second type of spasticity, soft spasticity, defining regions wherein any joint movement is somewhat difficult for the patient. In the model, this movement results in a penalty that is added to the node travel distance. Thus, while soft spasticity allows movement in a spastic region, it discourages it. This spasticity definition may be useful when analyzing completion of daily living tasks by individuals with mild spasticity impediments. Using soft spasticity, a penalty weighting can be appropriately assigned so that a path through a spastic region will be selected only when such movement is necessary for task completion. More generally, by varying the penalty weightings for different joints, the algorithm can be used to identify alternative paths for the same reach task. Additionally, the total spasticity travel distance penalty calculated for a path can help classify its difficulty for a patient. Research [18], [19] shows a linear relationship between force applied to move a spastic joint during slow stretch and the angular deviation from a threshold. Accordingly, the algorithm uses a linear penalty model for soft spasticity. The penalty is proportional to the average force involved in the movement and the angular distance moved. As the force is proportional to angle, the penalty equation is expressed as factors of angles and a conversion constant:

$$\text{Penalty} = F \times D \times C \tag{1}$$

F: average force angle is the average of the initial and final angles in the soft spasticity region; these angles are expressed as non-negative delta values relative to the edge of the region, and are based on the initial and final angles of a movement.

D: angular distance is the absolute value of the difference between the final and initial angles in the region.

C: constant, typically individual- and task-specific, that converts the product of the average force angle and angular distance to a travel distance penalty with units of centimeters.

An attractive conceptual property of this soft spasticity penalty definition is additivity, i.e., the penalty for a single movement of $x^o + y^o$ into the spastic region would be the same as the sum of the penalties for first moving x^o and subsequently an additional y^o:

$$\begin{aligned}(\text{initial } 0^o + \text{final } x^o + y^o) \times \text{angular distance } x^o + y^o = \\ (\text{initial } 0^o + \text{final } x^o) \times \text{angular distance } x^o + \\ (\text{initial } x^o + \text{final } x^o + y^o) \times \text{angular distance } y^o\end{aligned} \tag{2}$$

III. Results

The algorithm was evaluated by simulating a shoulder reach task both with and without synergy and spasticity. The task started with the arm straight down and ended with the hand outstretched in front of the body (see Fig. 10). Synergy data from [4] were used to generate the synergy matrices corresponding to different subjects from that study.

The most basic simulation had no synergy and no spasticity, resulting in the following graph of joint values in Fig. 3 and X, Y, and Z hand positions shown as overlays in Fig. 4.

Because there was no spasticity, the path taken approximates a straight line so as to minimize distance, utilizing shoulder flexion in combination with elbow flexion and wrist flexion. Compared to subsequent simulations, this one had the shortest travel distance, 119 cm. As the distance between the starting hand position and the target was approximately 116 cm, given the assumed arm length of 81 cm, the algorithm's solution was nearly as efficient as possible despite the fixed joint command increment of 3º.

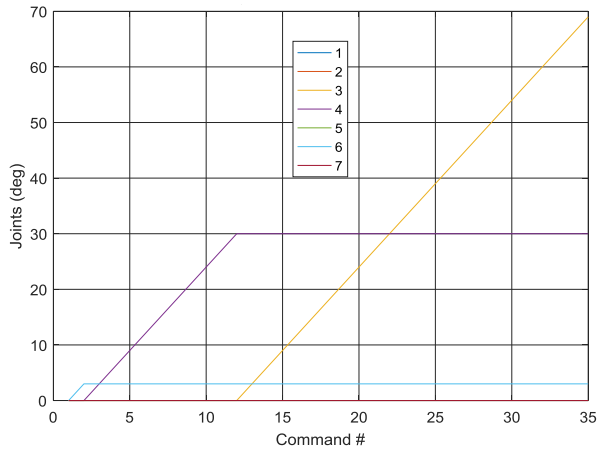

Fig. 3. Joint path with no synergy and no spasticity

The algorithm was also run using the synergy matrix of subject 3 from [4]. This subject was chosen because his synergy matrix score of 11, calculated using the method the paper devised for scoring synergy, most closely matched the average healthy score of 11.63. The resulting joint path is shown in Fig. 5; the XYZ path was omitted because it is similar to that in Fig. 4. The algorithm still mainly uses joints 3 and 4. However, synergy with the wrist (rather than joint 5 commands) causes large amounts of pronation and smaller quantities of wrist flexion and wrist ulnar deviation. Although the final joint angles differ from those in Fig. 3, the final XYZ hand position is also near the target. The dip in joint angle 5 is a result of a -5 command to counteract the synergy, ultimately allowing joints 6 and 7 to go towards the target instead of moving away from it. Because the synergy causes deviations from the straight line, the travel distance is slightly increased to 120 cm. The X coordinate remains close to 0º as movement in this direction is away from the target, is generally avoidable, and is not caused by this subject's synergy.

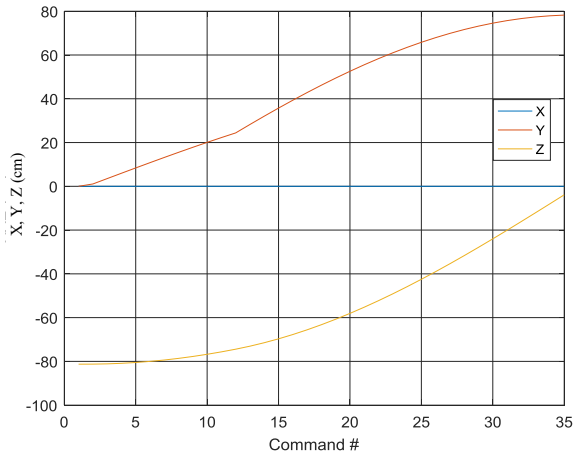

Fig. 4. Hand XYZ path with no synergy and no spasticity

The algorithm was also run using stroke subject 19's matrix, which had a synergy matrix score of 6, exhibiting slightly more synergy than the average (7) for all stroke subjects. The resulting joint and XYZ paths in Figs. 6 and 7, respectively, match the expectation that excessive and deleterious synergy would impair movement. It caused the arm to unnecessarily move off the X axis. The switchbacks seen in the joint graph reveal that commands were taken to counteract the synergistic response. Total travel distance was 132 cm.

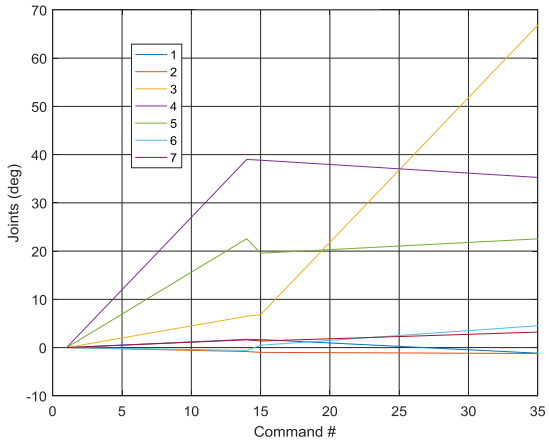

Fig. 5. Joint path with healthy synergy and no spasticity

To verify soft spasticity, two simulations were run with soft spasticity and no synergy. The first involved penalizing joint 4, elbow flexion. Based on data from [18], upper and lower spasticity thresholds were set at 90º, so any deviation from 90º incurred penalties. As Fig. 8 shows, because the task could be completed without moving the elbow, joint 4 is not moved. Instead, more shoulder flexion and wrist flexion are used to compensate, an example of the algorithm's alternative path generation capability.

The second simulation implemented soft spasticity on joint 3, shoulder flexion, with an upper threshold set at 0º. In contrast to the previous simulation, shoulder flexion is crucial to completing the task, so the solution illustrated in Fig. 9 utilizes shoulder flexion but minimizes it, compensating with more elbow flexion.

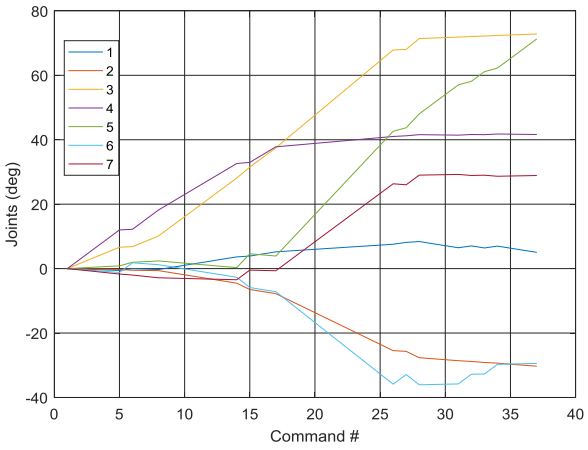

Fig. 6. Joint path with stroke-induced synergy and no spasticity

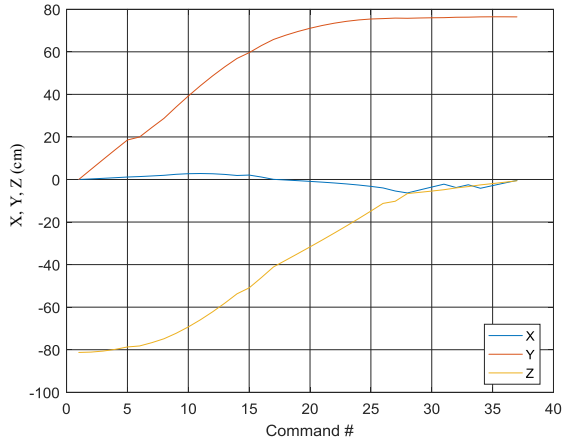

Fig. 7. Hand XYZ path with stroke-induced synergy and no spasticity

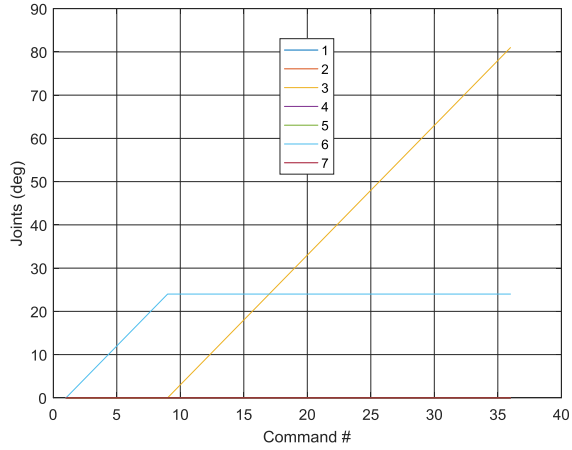

Fig. 8. Joint path with no synergy and joint 4 soft spasticity

Fig. 10 shows the starting arm configuration and the positioning of the arm halfway and at the end of the reach task when subject 3's healthy synergy was modeled. The major axes, in black, correspond to the starting frame, while axes of the hand are in color.

Lastly, some simulations were run with both synergy and spasticity. The results were consistent with previous simulations in that the spastic region was largely avoided, although synergistic interactions resulted in a small amount of movement in the region.

The set of all simulation results matched expectations, with spastic joint movements being avoided, synergy increasing solution travel distance, and joint commands counteracting synergy.

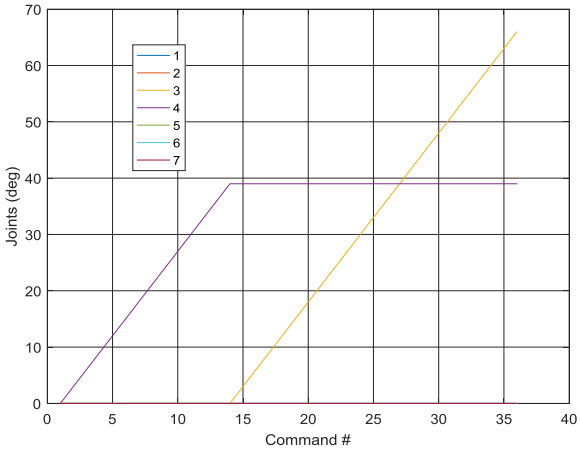

Fig. 9. Joint path with no synergy and joint 3 soft spasticity

IV. CONCLUSIONS

Based on the simulation results, the algorithm successfully minimizes travel distance while taking into account spasticity and synergy. Although the algorithm provides a general framework for generating reach task solutions, more data and analysis could help refine the modeling methods: the robotic arm model, the scoring criteria, synergy, and spasticity. Firstly, the seven DOF arm model, while standard, is a purely kinematic model. A more physical model that takes into account dynamics, namely gravity, the mass of the arm, and joint torques [20] would have greater fidelity. Secondly, although the current scoring metric minimizes distance to target, research shows that multiple factors may be involved, based on the arm position [21]. By choosing appropriate algorithm weights that could vary throughout the task, the scoring could be updated to take these factors into account. Thirdly, while the current synergy model is linear and joint-state-independent, a more sophisticated model could use higher-order polynomials and allow state-dependence [22]. For example, a set of synergy matrices could be used during the task, with the matrix selection depending on the joint values. Lastly, the flexibility for soft spasticity weighting suggests that future research should be conducted to further validate the linear model and to determine appropriate individualized weighting.

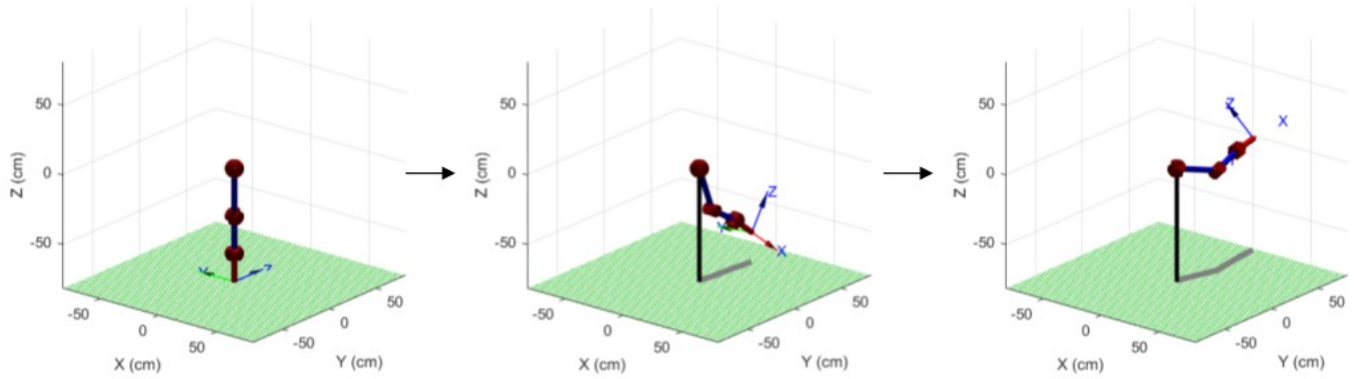

Fig. 10. Starting, halfway, and ending arm configurations for healthy synergy

REFERENCES

[1] Benjamin, Emelia J., Michael J. Blaha, Stephanie E. Chiuve, Mary Cushman, Sandeep R. Das, Rajat Deo, et al. "Heart disease and stroke statistics 2017 update: a report from the American Heart Association." Circulation, vol. 135, no. 10, 25 Jan. 2017, pp. e146–e603.

[2] Raghavan, Preeti. "Upper limb motor impairment post stroke." Physical Medicine and Rehabilitation Clinics of North America 26.4, 2015, pp. 599–610.

[3] Sukal, Theresa M., Michael D. Ellis, and Julius P. A. Dewald, "Shoulder abduction-induced reductions in reaching work area following hemiparetic stroke: neuroscientific implications." Experimental Brain Research, vol. 183, no. 2, 20 July 2007, pp. 215–223.

[4] Al-Refai, Aimen Hamid. "Objectively characterized linear model of stroke induced joint synergies in relation to clinical measures." EScholarship University of California, 7 Dec. 2012, escholarship.org/uc/item/38x1v76f.

[5] Sommerfeld, Disa K., Elsy U.-B Eek, Anna-Karin K. Svensson, Lotta W. Holmqvist, and Magnus H. Von Arbin. "Spasticity after stroke: its occurrence and association with motor impairments and activity limitations." Stroke, vol. 35, 29 Dec. 2003, pp. 134–139.

[6] Urban, Peter P., Thomas Wolf, Michael Uebele, Jürgen J. Marx, Thomas Vogt, Peter Stoeter, et al. "Occurrence and clinical predictors of spasticity after ischemic stroke." Stroke, vol. 41, no. 9, 30 Aug. 2010, pp. 2016–2020.

[7] Dipietro, L., H. I. Krebs, S. E. Fasoli, B. T. Volpe, J. Stein, C. Bever, et al. "Changing motor synergies in chronic stroke." Journal of Neurophysiology 98.2, 2007, pp. 757-68.

[8] Ada, Louise, Simone Dorsch, and Colleen G. Canning. "Strengthening interventions increase strength and improve activity after stroke: a systematic review." Australian Journal of Physiotherapy 52.4, 2006, pp. 241-48.

[9] Gracies, J. M. "Pathophysiology of impairment in patients with spasticity and use of stretch as a treatment of spastic hypertonia." Physical Medicine and Rehabilitation Clinics of North America 12.4, 2001, pp. 747-68.

[10] Bayona, Nestor A., Jamie Bitensky, Katherine Salter, and Robert Teasell. "The role of task-specific training in rehabilitation therapies." Topics in Stroke Rehabilitation 12.3, 2005, pp. 58-65.

[11] Faria, Ana L., Andreia Andrade, Luisa Soares, and Sergi Bermudez I Badia. "Benefits of virtual reality based cognitive rehabilitation through simulated activities of daily living: a randomized controlled trial with stroke patients." Journal of NeuroEngineering and Rehabilitation 13.1, 2016, pp. 1-12.

[12] Laver, Kate, Stacey George, Susie Thomas, Judith E. Deutsch, and Maria Crotty. "Virtual reality for stroke rehabilitation." Stroke 43.2, 2012, pp. E20-21.

[13] Rosen, Jacob, and Joel C. Perry. "Upper limb powered exoskeleton." International Journal of Humanoid Robotics, 2007, pp. 529-48.

[14] Byl, Nancy N., Gary M. Abrams, Erica Pitsch, Irina Fedulow, Hyunchul Kim, and Matt Simkins. "Chronic stroke survivors achieve comparable outcomes following virtual task specific repetitive training guided by a wearable robotic orthosis (UL-EXO7) and actual task specific repetitive training guided by a physical therapist." Journal of Hand Therapy 26.4, 2013, pp. 343-52.

[15] Corke, Peter I. Robotics, Vision and Control: Fundamental Algorithms in Matlab. Berlin: Springer, 2013.

[16] Asfour, T., and R. Dillmann. "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem." Proceedings 2003 IEEE/RSJ Iinternational Conference on Intelligent Robots and Systems (IROS 2003), 2003, pp. 1407-1412.

[17] Luttgens, Kathryn, Nancy Hamilton, and Helga Deutsch. Kinesiology: Scientific Basis of Human Motion. Maddison: Brown & Benchmark, 1997.

[18] Park, Hyung-Soon, Jonghyun Kim, and D. L. Damiano. "Haptic recreation of elbow spasticity." 2011 IEEE International Conference on Rehabilitation Robotics, 2011, pp. 1-6.

[19] Zakaria, Noor Ayuni Che, Takashi Komeda, Cheng Yee Low, Fazah Akhtar Hanapiah, and Kaoru Inoue. "Spasticity mathematical modelling in compliance with modified Ashworth scale and modified Tardieu scales." 2015 15th International Conference on Control, Automation and Systems (ICCAS), 2015, pp 1893-1897.

[20] Perry, Joel C., Janet M. Powell, and Jacob Rosen. "Isotropy of an upper limb exoskeleton and the kinematics and dynamics of the human arm." Applied Bionics and Biomechanics 6.2, 2009, pp. 175-91.

[21] Li, Zhi, Dejan Milutinovic, and Jacob Rosen. "Spatial map of synthesized criteria for the redundancy resolution of human arm movements." IEEE Transactions on Neural Systems and Rehabilitation Engineering 23.6, 2015, pp. 1020-030.

[22] Simkins, Matt, Anne Burleigh Jacobs, Nancy Byl, and Jacob Rosen. "Stroke-induced synergistic phase shifting and its possible implications for recovery mechanisms." Experimental Brain Research 232.11, 2014, pp. 3489-499.