BIOMIMETIC CONTROL OF SPACEBORNE MANIPULATOR FOR DEBRIS REMOVAL AND ON-ORBIT SERVICING

Virtual Conference 19–23 October 2020

Collins Ogundipe[1], Alex Ellery[2]

[1]Department of Mechanical & Aerospace Engineering, Carleton University, 1125 Colonel by Drive, Ottawa, K1S 5B6, Canada, E-mail: collinsogundipe@cmail.carleton.ca

[2]Department of Mechanical & Aerospace Engineering, Carleton University, 1125 Colonel by Drive, Ottawa, K1S 5B6, Canada, E-mail: alexellery@cunet.carleton.ca

ABSTRACT

A traditional approach to space debris mitigation is the removal and de-orbiting of debris. We propose a novel approach which involves in-situ resource utilization of salvaged debris for re-use as a development of on-orbit services. This could be regarded as a more sustainable approach to space debris control. Only robotic manipulation is flexible enough to deal with both large and small debris unlike harpoons and nets which generate complex uncontrollable dynamic interactions between the robotic free-flyer, the target and the flexible umbilical connecting the two. This favors space debris mitigation through the deployment of free-flyer spacecraft mounted with dexterous manipulators which provide controlled interaction with the target [1]. Robot manipulators have been the workhorse of industrial applications for a range of tasks where precision positioning is required including machining, welding, sanding, spraying and assembly. Debris mitigation will require robust robotic grasping. Currently, most approaches to tactility in robotic grasping rely on either significant processing resources or soft robotics. There is little doubt that robust and adaptable robotic grasping would be a boon in both orbital debris removal and on-orbit servicing [2].

We propose here a novel bio-inspired error-learning approach to manipulator control that specifically addresses the requirement for adaptability and compliance to a range of dynamic tasks under dynamic environmental conditions. We propose to solve this problem by accommodating a different payload inputs and a force control model in the error learning algorithm. To avoid unreliable predictions for robustness, the error learning models of the robotic manipulator were trained on data accounting for possible perturbations and different payloads. We have introduced a task specific model that is able to learn from its errors (make error predictions) under different payloads and varying environmental dynamics, to accommodate space debris removal of varying sizes. This is entirely different to model predictive control. For now, we present here, a two-layer approach to grasping: (i) hybrid force/impedance control through feedback, which is the traditional approach - but delays in the feedback cycle can generate instabilities; (ii) the addition of a feedforward predictive capability to partially circumvents this problem by emulating the function of the human cerebellum (effectively, a Kalman filter-based neural network). Essentially, we have developed appropriate neural net models for varying sizes of payloads and dynamics – using models of the 7 degrees-of-freedom Barrett Arm.

1 INTRODUCTION

For the control of spaceborne manipulators required for grappling space debris targets, modeling the force control to achieve adaptable and compliant behavior is a major obstacle hindering the deployment of robotic free-flyers for such a critical task. Previously explored avenues for achieving reactive and compliant behavior, for manipulators in general, is the representation of motions through acceleration-based control. Acceleration policies can typically support the realization of real-time, adaptive, and complaint robots. However, tracking acceleration is only feasible when we do have an accurate model of the inverse dynamics of the system. This poses a challenging task as we do not know how to accurately change the robot's state in dynamic environments or under dynamic tasks. With no precise or single-task model of the systems dynamics, we usually resort to tracking desired trajectories by integrating the acceleration policy while deploying feedback control to reject the modeling errors of the dynamics. This in turn presents a trade-off between the compliancy and reactiveness of our controller against accuracy, as effort is being made to tune the gains (higher) to achieve good tracking performance. For the purpose of space debris removal (salvaging) and/or on-orbit servicing, robotic free-flyers' regimes of operations will include free-space motion, transitional contact dynamics,

transitional passivation of target, and some servicing operations such as peg-in-hole task. Therefore, it will be required to perform mixed mode position and force control covering these tasks, but with the aim to achieve reactive and compliant behaviour. In order to achieve this, it has been proposed that models should instead be learned online using streams of sensory data, in a process where it can infer the characteristics of its structure and environment. Considerable effort has been put into developing machine learning methods that can learn and improve inverse dynamics model [3-6]. Online learning has been the focus in these settings because when considering motions with object interactions, learning one global model becomes very challenging, if not impossible, since the model must be a function of contact and payload signals. To approach the issue of global/dynamic model, learning task-specific (error) models has been proposed in the past [7-10], such that the overall global problem is simplified into two subproblems – (1) finding a task-specific inverse dynamics model and (2) detecting which task model to use. This permits to iterate the collection of data specific to a task, learn an error model, and then apply the learned model during the required task execution. However, a key difficulty that has been encountered is the computationally efficient learning of models that are data-efficient as possible, such that only few iterations are required while achieving consistent convergence in the error model learning. One major challenge to (online) inverse dynamics learning is computational efficiency, especially more so when it comes to space application. We seek to address this using predictive feedforward approach, in a pre-learned fashion, ensuring a more compliant and reactive robot. Our take on this is that pre-learned input-output models are computationally efficient compared with analytical models – the latter require exact knowledge of parameters (commonest sources of errors which include payload variation) and require computation time. Learned models reduce computation by storing model in memory. Increased gains increase the control effort and the tendency to instability.

2 CHALLENGES WITH PAST LEARNING APPROACHES

One major challenge to online inverse dynamics learning is computational efficiency, especially more so when it comes to space application. Predicting with learned models needs to be feasible within the real-time constraints of the systems consuming torque commands.

Another major challenge to the traditional inverse dynamics learning is that increasing the gains of the feedback control in an attempt to circumvent extreme cases of bad tracking does not only affect compliancy; but also, no useful data of inverse dynamics model learning would be generated because the collected data points will be slightly off the desired trajectory for which we aim to deduce the inverse dynamics model. An alternative approach was presented in a recent work [11], where a direct loss function was employed to minimize the error between the desired and actual accelerations, to learn the feedback terms online. In this study [11], the idea was that feedback control could be viewed as an online technique to compensate for errors in a given a priori inverse dynamics model. The feedback terms compensate for errors between what a given model predicts (estimates) and what is actually obtained. As such, they naturally act as a convenient source of training data. In a further study [12], it was shown how the direct loss on accelerations as presented in [11] can be transformed into a loss on inverse dynamics torques, measured at desired accelerations as against at actual accelerations. It was shown in [12] how the combination of these two data sources (training data points measured at actual acceleration and the training data source measured at desired acceleration) leads to more consistent convergence and often faster learning convergence of a task-specific inverse dynamics learning process. However, this study was only carried out for a single task or a task-specific case. A major limitation of the study [12] was that the input of the error model does not account for any information of the payload term, and such a system would lead to data ambiguities (not being able to separate payload contained/driven data from non-payload driven data). As cited as example, picking up a tool from a position, placing it somewhere else and repeating the pickup without the tool could not be expressed by a single task-specific inverse dynamics model for now. Furthermore, in the study [12], it was not analyzed how the system performs when it is strongly perturbed. And this could result in undesirable predictions since the error model has not been trained on any data for that input space. This has been a general problem for task specific models.

3 BIOMIMETIC APPROACHES TO ROBOT MANIPULATION

Compliant manipulation is a major constraint for grappling in robotic manipulators. The goal is to reproduce human level manipulation capabilities – it would be necessary then to examine human

manipulation from a biomimetic standpoint. The problem in introducing robots into the wider world has not been their intelligence but their ability to physically interact with the world. Robust adaptive manipulation is the key to converting space junk into salvageable assets for re-use. Key facets to bio-inspired approaches to engineering are robustness and adaptability [13]. It is believed that biomimetic approaches could provide this capability. A tutorial review of this approach to robotic manipulation has been presented, emphasising the central role played by manipulator control systems [14].

Sensorimotor control is the primary function of the brain for which several strategies are employed [15]: (i) sensorimotor planning, learning and control; (ii) optimal feedback control, (iii) impedance control, (iv) predictive control, and (v) Bayesian inferencing. According to [14], "In the human brain, the primary motor cortex and supplementary motor area encode adaptation of kinematic-dynamic transformations of movements. Voluntary movement requires three main computational processes to be implemented in the brain: (i) determination of cartesian trajectory in visual coordinates; (ii) transformation of visual coordinates into body coordinates in which proprioceptive feedback occurs (within the association cortex); (iii) the cartesian trajectory in body coordinates (joint coordinates) θ^d is converted into the generation of motor commands τ (within motor cortex) to the muscles through the spinal cord. Internal models are used as neural models of aspects of the sensorimotor loop including interaction with the environment to predict and track motor behaviour. The primary motor cortex (M1) implements inverse models that convert desired end effector cartesian trajectories into patterns of muscle contractions at the joints (output), i.e. motor commands." These coordinate transformations between external world coordinates to joint/muscle coordinates appear to be implemented between M1 and the ventral premotor cortex (PMV) [16]. The first mapping that must be achieved is the nonlinear transformation of task (cartesian) coordinates of the end effector q in terms of joint coordinates θ:

$$q = f(\theta)$$

where f(θ) = 4x4 Denavit-Hartenburg matrix. This can be differentiated to get the cartesian velocities in relation to joint velocities through the Jacobian matrix J(θ):

$$\dot{q} = J(\theta)\dot{\theta}$$

where J(θ) is 6-by-n Jacobian matrix for n degrees of freedom. From virtual work reasoning, the transpose of the Jacobian relates joint torques τ to the cartesian end effector forces F: $\tau = J^T F$

If the manipulator is kinematically redundant (i.e. n>6), the Moore-Penrose pseudoinverse is the generalised inverse: $J^* = J^T(JJ^T)^{-1}$

The inverse dynamic representation for a robotic manipulator is given by the Euler-Lagrange equations illustrating the output torque τ required to attain the observable kinematic state of the manipulator joints $(\theta, \dot{\theta}, \ddot{\theta})^T$:

$$\tau = D(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right) + G(\theta) + J^T F$$

where D(θ) is the inertia parameter of the manipulator; $C\left(\theta, \dot{\theta}\right)$ is the coriolis and centrifugal parameter of the manipulator; G(θ) is the gravitational parameter of the manipulator (this term vanishes for space manipulators); F is the external force at the manipulator end effector; and J is the Jacobian matrix. The adaptive finite impulse response (FIR) filter may be used to approximate inverse dynamic model of a process through mean square error minimization [17]. Numerous regions of the brain project into motor area M1 providing feedback signals – primary somatosensory cortex, posterior area 5 and from the thalamus via the cerebellum. In feedback control, the actual trajectory is compared with the desired trajectory, and by this means defining the tracking error. This error is fed back to the motor command system to allow adjustments to decrease this error. Control systems utilize inverse models to calculate the desired motor action necessary to achieve the desired effects on the environment (such as a desired trajectory). The feedback controller calculates the motor command based on the error between the desired and estimated states. The motor command is the sum of feedback controller command and the inverse model output. Inverse internal modelling of the kinematics and dynamics of motion is like adaptive sliding control [18]. The internal model represents an observer and essentially represents the reference model used in adaptive controllers. An inverse model may be produced by inverting a forward model neural network representing the causal process of the plant [19]. Forward models define the causative relationship between the torque inputs and the outputs states of the motor trajectory (position/velocity) and the sensory states given these estimated states. The parietal cortex is concerned with visual control of hand movements - it requires ~135-290 ms to process visual feedback [14]. It calculates the error between the desired Cartesian position and the current Cartesian position, the latter calculated from proprioceptive feedback measurements from muscle spindles [20]. This needs

an efference copy of the motor commands to generate a feedforward compensation model. An efference copy (corollary discharge) of these motor commands is passed to an emulator that models the input-output behaviour of the musculoskeletal system. A hierarchical neural network model can emulate the function of the motor cortex [21]. There is an error between the actual motor patterns θ (and $\dot{\theta}$) measured by proprioceptors and the commanded motor patterns τ from the motor cortex which is fed back as $\theta^d - \theta$ with a time delay of 40-60 ms [14]. A forward dynamics model of the musculoskeletal system resides within the spinocerebellum-magnocellular red nucleus system. The forward model receives feedback from the proprioceptors θ and an afferent copy of the motor command τ. Thus, the forward model takes motor command τ as input and outputs the predicted trajectory θ* [14]. The forward model predicts the movement θ* which is used in combination with motor command τ to calculate a predicted error $\theta^d - \theta^*$ which is transmitted to the motor cortex with a much shorter time delay of 10-20 ms [14]. The forward model predicts the sensory consequences of the motor commands. This top-down prediction is based on a statistical generative model of the causal structure of the world learned through input-output relations. In humans, this forward model of the musculoskeletal system has been learned since the earliest motor babbling that begins after birth [14]. An inverse dynamics model of the musculoskeletal system exists within the cerebrocerebellum-parvocellular red nucleus system – it does not receive sensory inputs. The inverse dynamics model has the desired trajectory θ^d as input from which it computes motor commands τ as output. The inverse dynamics model must learn to match the forward model to generate accurate motor commands τ in order to compensate for variable external forces [14]. The integral forward model paradigm places the forward model at the core of all perception-action processes – this is the basis for the integral forward model in which sensor and motor functions are fully integrated [22]. Forward models are utilized to make predictions that offer top-down expectations to incoming bottom-up sensory information. Disparity causes a prediction error that prompts refinement of expectations.

4 PREDICTIVE FEEDFORWARD CONTROL

We are proposing a novel bio-inspired error-learning approach that specifically addresses the requirement for adaptability and compliance to a range of dynamic tasks under dynamic environmental conditions. If we could successfully demonstrate this for a terrestrial manipulator, the idea is to adopt/modify the approach in a free-flyer concept for the removal of space debris of varying sizes, offering a salvage solution that is robust to any orbital band. In effect, we are proposing a three-level control strategy based on biomimetic forward models for predictive estimation, traditional feedback control and biomimetic muscle-like preflexes. We place emphasis on bio-inspired forward modelling suggesting that all roads lead to this solution for robust and adaptive manipulator control. This promises robust and adaptive manipulation for complex tasks in salvaging space debris.

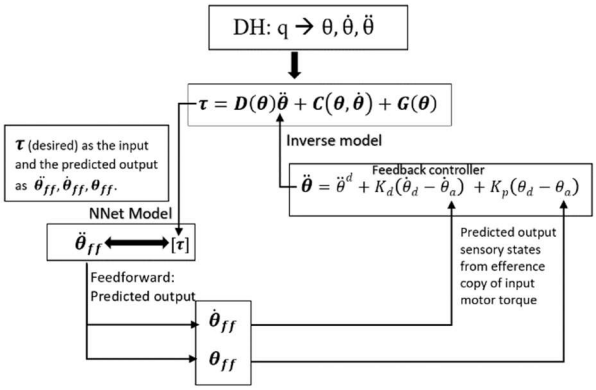

Figure 1: The predictive forward model scheme. The neural network ("NNet") model is trained using data from experimental teaching mode.

A forward model has been implemented and trained as a neural network approximator using some trajectory dataset relating the output torque τ gotten from the kinematic state of the manipulator joints $(\theta, \dot{\theta}, \ddot{\theta})^T$ in an experimental teaching mode. As shown in Fig. 1, the trained neural network model will hence be able to take the analytically calculated torque (efference copy of input motor commands) as its input, while the output of the neural network will be the predicted trajectory output $(\theta_{ff},\ \dot{\theta}_{ff},\ \ddot{\theta}_{ff})^T$. This system incorporates an inverse model with an adaptive feedforward component, i.e., it comprises a feedforward and feedback loop. The forward internal model of the system dynamics allows prediction of the motor system state similarly to a Kalman filter. The output of the feedback controller is used as an error signal to train the feedforward controller which models the inverse dynamics. The feedback controller is essentially a self-tuning adaptive controller/sliding

controller while the feedforward controller employs a gradient descent to minimize the error.

Significant delays in neural feedback signals from sensors make pure feedback strategies improbable, so predictive feedforward control is necessary with the feedback being used to correct the trajectory. Two copies of a motor command are generated by the inverse model, the efference copy being passed to the forward model to simulate the expected (desired) sensory consequences which are then compared with actual sensory feedback. The forward predictive model is essential for skilled motor behaviour – it models how the motor systems respond to motor commands. In the forward model, motor commands are input to be converted into their sensory consequences as the output – they model the causative relationship between input actions and their effects on the environment as measured by the sensors. The forward internal model behaves as a simulator of the body and its interface with the environment, i.e. it represents a predictor.

The forward dynamic model of a robotic manipulator is given by (for the sensory joint acceleration rate):

$$\ddot{\theta} = D^{-1}(\theta)[\tau - C(\theta, \dot{\theta}) - G(\theta)]$$

Joint acceleration $\ddot{\theta}$ could be integrated to yield joint velocity rate $\dot{\theta}$ and joint rotation θ as the predicted output sensory states for torque input τ. The predictive forward dynamic model mimics the body's muscular nature which produces a predicted trajectory output from an efference copy of input motor commands [23]. Feedforward control consequently uses a model of the plant process to predict its response to disturbances so as to compensate for time delays [24]. The predicted trajectory output may be supplied to the input of the feedback controller to compensate for time delays – Figure 1. Forward models adapt 7.5 times more quickly than inverse models alone [25]. This forward predictive control scheme has been recommended as a model of cerebellar learning from proprioceptive feedback from muscle spindles and Golgi apparatus which measure muscle stretch. The forward model may be executed as a neural network function approximator to the forward dynamics. This could be represented as a look-up table weights learned from input-output pairs of visuomotor training data (in an offline, pre-trained fashion).

The forward model utilizes an efference copy of the motor command as input to cancel the sensory effects of movement. The same process cancels the effects of self-motion on sensation to distinguish from environmental effects (e.g. self-tickling). For each/any forward model, there is a paired inverse model to generate the required motor command for that context dictated by sensory signals. In motor control, the full internal model encompasses a paired set of forward and inverse models involving two network pathways through an inverse model and forward model respectively, with the latter working as a predictor. The feedback controller transforms desired effects into motor commands while the feedforward predictor transforms motor commands into anticipated sensory consequences.

Multiple paired forward and inverse models would be crucial to handle the large number of kinematic-dynamic situations that could possibly arise.

5 RESULTS - PREDICTIVE FEEDFORWARD MODEL

We present here the results of the predicted feedforward model. The significance of the result is to demonstrate how we have developed neural network models which are capable of predicting (to a high degree of accuracy) forward trajectory variables (θ_{ff}, $\dot{\theta}_{ff}$, $\ddot{\theta}_{ff}$) from an efference copy of the torque, as shown in Figure 1. It means the models are poised to cancel the sensory effects of the arm movement, providing anticipated sensory consequences from the motor command. With this, instabilities that could arise in delays when using traditional feedback cycle has been partially circumvented. This is akin to how the human cerebellum functions as discussed in Section 3.

Here, we adopt a dataset publicly made available by [26], where the WAM Barrett Arm was taken by the end-effector and guided along several trajectories in a teaching mode. During the imagined motion, the joint trajectories ($\theta, \dot{\theta}, \ddot{\theta}$) were sampled from the robot and the corresponding motor torques (τ) measured for each data point. The dataset has a total of 12,000 samples. For the 7 degree-of-freedom Barrett arm, 7 motor torques were measured, along with 21 joint trajectory variables representing seven joint angles (θ), seven joint velocities ($\dot{\theta}$) and seven joint accelerations ($\ddot{\theta}$).

Deep learning neural network models were developed, and we learned the forward dynamics model by using the joint torques as input and the joint trajectories as targets. The deep learning algorithm developed is of multiple-target prediction, under multiple output regression. For a feature vector *x*, we aim to predict accurately a vector of responses *y* using a function *h(x)*:

$$\boldsymbol{x} = (x_1, x_2, \dots, x_p) \overset{h(x)}{\Longrightarrow} \boldsymbol{y} = (y_1, y_2, \dots, y_m)$$

Some of the main challenges is the appropriate modeling of target dependencies between targets $y_1, y_2, \dots, y_m$, and having a multitude of multivariate loss functions defined over the output vector, $\mathcal{L}(y, h(x))$. The independent features were used to train each set of targets grouped separately by the joint angles, joint velocities, and the joint accelerations. Meaning three different models with different hyperparameters were trained (learned), in an attempt to manage target dependencies. The first model relates the 7 joint torques as input to the 7 joint angles as targets; the second model was learned between the 7 joint torques as input and the 7 joint velocities as targets, while the third model learned the relationship between the 7 joint torques as input and multiple-target prediction of the 7 joint accelerations as the output. The neural network algorithm was written in Python, while some *keras* and *scikit-learn* packages were used. Below, we present a table of results comparing the predicted feedforward joint angles to some known joint angles separated as test set from the dataset. A similar table of results is shown for the joint velocity targets, for a sample data point.

Table 1: Prediction Accuracy of Joint Angles

Joint Number	Joint Angle (rad) ***Test Set***	Joint Angle (rad) ***Predicted***	Accuracy (%)
1	0.0734	0.0669	91.1
2	0.5349	0.5777	92.6
3	0.0126	0.0110	87.3
4	1.5386	1.5781	97.4
5	0.2453	0.2315	94.4
6	0.0555	0.0528	95.2
7	0.0994	0.0957	96.3

Table 2: Prediction Accuracy of Joint Velocities

Joint Number	Joint Velocity (rad/s) ***Test Set***	Joint Velocity (rad/s) ***Predicted***	Accuracy (%)
1	0.0445	0.0403	90.5
2	-0.1626	-0.1533	94.3
3	-0.1239	-0.1106	89.3
4	-0.0793	-0.0812	97.7
5	-0.0153	-0.0147	96.0
6	0.0459	0.0435	94.7
7	-0.03104	-0.0296	95.3

With the prediction accuracy of between 87-98%, it can be seen that our models compared favorably to the expected predictive feedforward, and it is able to provide anticipatory sensory consequences from the motor command. Instabilities that could arise in delays from using traditional feedback can be partially circumvented, an important step in our biomimetic approach to developing Kalman filter-based neural network models for varying sizes of payloads and dynamics.

6 CONCLUSION AND FUTURE WORK

With the predictive neural network models showing promising results, part of the future work is to use the efference (analytical dynamics) torque on the models to generate the feedforward trajectory variables fed into the feedback controller – Figure 1. The resulting torque will be used to drive the robot and comparative analysis will be made between the response of the robot in comparison to what is obtained in the traditional approach. Along with this predictive feedforward model (which will incorporate hybrid force/impedance control), we intend to implement a software-based viscoelastic response to the electric motors to emulates muscle/tendon behaviour toward providing robustness to grappling errors. This shall equally involve the implementation of neural net models embedded in a Bayesian gating algorithm that demonstrates the proper selection of the right neural net model for varying sizes of payloads – using terrestrial Barrett Arm. Following an anticipated success, we shall incorporate this into our knowledge of robotic free-flyers dynamics to demonstrate viability for space application for debris removal and on-orbit servicing.

References

[1] Liou J-C, Johnson N, Hill N (2010) "*Controlling the growth of future LEO debris populations with active debris removal*" Acta Astronautica 66, 648–653

[2] Castronuovo M (2011) "*Active space debris removal—a preliminary mission analysis and design*"

Acta Astronautica 69, 848–859

[3] S. Vijayakumar and S. Schaal (2000), "*Locally weighted projection regression: Incremental real time learning in high dimensional space*," in Proc. the International Conference on Machine Learning (ICML), pp. 1079–1086.

[4] D. Nguyen-Tuong, J. R. Peters, and M. Seeger (2008), "*Local Gaussian process regression for real time online model learning*," in Proc. Neural Information Processing Systems (NIPS), pp. 1193–1200.

[5] A. Gijsberts and G. Metta (2013), "*Real-time model learning using incremental sparse spectrum Gaussian process regression*," Neural Networks, vol. 41, pp. 59–69.

[6] F. Meier, P. Hennig, and S. Schaal (2014), "*Incremental Local Gaussian Regression*," in Proc. Neural Information Processing Systems (NIPS).

[7] L. Jamone, B. Damas, and J. Santos-Victor (2014), "*Incremental learning of context-dependent dynamic internal models for robot control*," in Proc. of the IEEE International Symposium on Intelligent Control (ISIC).

[8] M. Toussaint and S. Vijayakumar (2005), "*Learning discontinuities with products-of-sigmoids for switching between local models*," in Proc. of the International Conference on Machine Learning (ICML).

[9] G. Petkos, M. Toussaint, and S. Vijayakumar (2006), "*Learning multiple models of non-linear dynamics for control under varying contexts*," in International Conference on Artificial Neural Networks. Springer, pp. 898–907.

[10] D. M. Wolpert and M. Kawato (1998), "*Multiple paired forward and inverse models for motor control*," Neural networks, vol. 11, no. 7.

[11] N. Ratliff, F. Meier, D. Kappler, and S. Schaal (2016), "*Doomed: Direct online optimization of modeling errors in dynamics*," Big Data, vol. 4, no. 4, pp. 253–268.

[12] D. Kappler, F. Meier, N. Ratliff, and S. Schaal (2017), "*A New Data Source for Inverse Dynamics Learning*," Proc of. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[13] Vincent J, Bogatyreva O, Bogatyrev N, Bowyer A, Pahl A-K (2006) "*Biomimetics: its practice and theory*" J Royal Society Interface 3, 471-482

[14] Ellery A. (2020) "*Tutorial review of bio-inspired approaches to robotic manipulation for space debris salvage*" Biomimetics Journal 12 (5), E19, 2020.

[15] Franklin D, Wolpert D (2011) "*Computational mechanisms of sensorimotor control*" Neuron 72, 425-442

[16] Kakei S, Hoffman D, Strick P (2003) "*Sensorimotor transformations in cortical motor areas*" Neuroscience Research 46, 1-10

[17] Widrow B (1986) "*Adaptive inverse control*" IFAC Adaptive Systems in Control & Signal Processing, 1-5

[18] Tin C & Poon C-S (2005) "*Internal models in sensorimotor integration: perspectives from adaptive control theory*" J Neural Engineering 2 (3), S147-S163

[19] Palade V, Puscasu G, Neagu D-C (2000) "*Neural network-based control by inverting neural models* "http://www.cs.ox.ac.uk/people/vasile.palade/papers/Palade_CEAI99.pdf

[20] Bullock D & Grossberg S (1998) "*Cortical networks for control of voluntary arm movements under variable force conditions*" Cerebral Cortex 8 (1), 48-62

[21] Kawato M, Furukawa K, Suzuki R (1987) "*Hierarchical neural network model for control and learning of voluntary movement*" Biological Cybernetics 57, 169-185

[22] Pickering M, Clark A (2014) "*Getting ahead: forward models and their place in cognitive architecture*" Trends in Cognitive Sciences 18 (9), 451-456

[23] Morasso P, Baratto L, Capra R, Spada G (1999) "*Internal models in the control of posture*" Neural Networks 12, 1173-1180

[24] Basso D, Belardinelli O (2006) "*Role of the feedforward paradigm in cognitive psychology*" Cognitive Processes 7, 73-88

[25] Flanagan J, Vetter P, Johansson R, Wolpert D (2003) "*Prediction precedes control in motor learning*" Current Biology 13, 146-150

[26] D. Nguyen-Tuong, M. Seeger, and J. R. Peters (2009), "*Model Learning with Local Gaussian Process Regression*," Advanced Robotics 23 2015–2034.