An Adaptive Modular Recurrent Cerebellum-Inspired Controller

Kiyan Maheri*, Alexander Lenz, Martin J Pearson

Bristol Robotics Laboratory, Bristol BS16 1QD, U.K.
km1768@bristol.ac.uk *
Alex.Lenz@brl.ac.uk
martin.pearson@brl.ac.uk

Abstract. Animals and robots face the common challenge of interacting with an unstructured environment. While animals excel and thrive in such environments, modern robotics struggles to effectively execute simple tasks. To help improve performance in the face of frequent changes in the mapping between action and outcome (change in context) we propose the Modular-RDC controller, a bio-inspired controller based on the Recurrent Decorrelation Control (RDC) architecture. The proposed controller consists of multiple modules, each containing a forward and inverse model pair. The combined output of all inverse models is used to control the plant, with the contribution of each inverse model determined by a responsibility factor. The controller is able to correctly identify the best module for the current context, enabling a significant reduction of 70.9% in control error for a context-switching plant. It is also shown that the controller results in a degree of generalization in control.

Keywords: cerebellum, adaptive control, bio-inspired, modular control, adaptive filter, context switching

1 Introduction

It can be observed in every day life that animals are accomplished at controlling their motor system when interacting with the environment. Traditional control methods have so far failed to deliver the same levels of competence for robots.

In an agent's interaction with the environment the mapping between actions and sensory feedback changes frequently. This mapping includes the dynamics of the agent's body, the environment and sensory processing by the agent. Changes in any of the three elements, perhaps resulting from picking up a new object or relying on a different sense for feedback, results in a change in the overall mapping. The mapping between actions and sensory feedback is represented in a single element known as the plant. This plant is able to switch between numerous different dynamics, herein referred to as contexts. This challenging control scenario is further compounded by the fact that new mapping is difficult to predict *a priori*. In each case animals show the ability to learn, retain and recall these mappings for control in a wide range of situations.

In mammalian Motor Control Systems (MCS) the sensory information is delayed and often very noisy.[1] This makes feedback control an unlikely mechanism for low-level biological control. However, delayed feedback can still be used for training adaptive controllers. In this work we propose a Controller (Modular-RDC) that is able to learn the control of multiple contexts and recall that control when needed. This architecture uses the adaptive feedforward RDC controller and develops a distributed control architecture around it.

It has been suggested that the MCS uses multiple adaptive control modules for storing motor skills.[2] According to this theory the MCS is able to identify and select the best modules for a given context. The ability to correctly identify and switch between pre-learned modules helps to overcome uncertainty and sudden changes in context. Here this theory is referred to as distributed control. This is in contrast to a single controller which has to relearn control every time the agent changes context. In the mammalian brain there is strong evidence to suggest that the Cerebellum is a likely candidate for the location of these parallel modules.[3, 4]

The Cerebellum is known as an important site of motor learning.[5] The Cerebellum does not generate the motor commands itself, but instead is engaged in fine-tuning movements.[6] The cerebellum has a well-characterized and regular structure, consisting of a large number of repeating sub-units known as microzones. It has been suggested that there exists a cerebellar function carried out by each microzone. In 1982 Fujita, working with the Marr-Albus model of the cerebellum, proposed that each microzone acts as an adaptive filter.[7] Adaptive filters are a powerful and versatile tool frequently used in signal processing.

Porrill et al. [8] proposed the Recurrent Decorrelation Control (RDC) model as a biologically plausible adaptive single-controller architecture (Figure 1). In this model the Cerebellar model takes in an efferent copy of the motor command and uses its output to modify the reference signal. The cerebellar model runs counter-current to a fixed feedforward controller, thus creating a recurrent loop between the two elements. The cerebellar model emulates a single microzone, which is modeled as an adaptive filter in accordance to the Fujita model. This controller has been used to emulate the Vestibulo-ocular reflex (VOR)[9] and control artificial muscle.[10] The controller is able to learn the control of a plant context. However, if the plant context changes the controller adapts and loses the previously learned control.

An important question in distributed control is how to assign responsibility among modules. Narendra and Balakrishnan [12] were the first to use multiple control modules to improve transient responses. In their architecture the plant is controlled by a set of fixed inverse models, each being coupled to a forward model. The inverse model paired with the best performing forward model is selected to drive the plant. Wolpert et al. proposed the MOSAIC architecture as a biologically plausible model for motor control.[2, 13] MOSAIC also makes use of forward models to select controllers. However, unlike Narendra's controller, MOSAIC combines the module outputs in a weighted sum, thus allowing for the control and learning of contexts to be shared among multiple controllers.

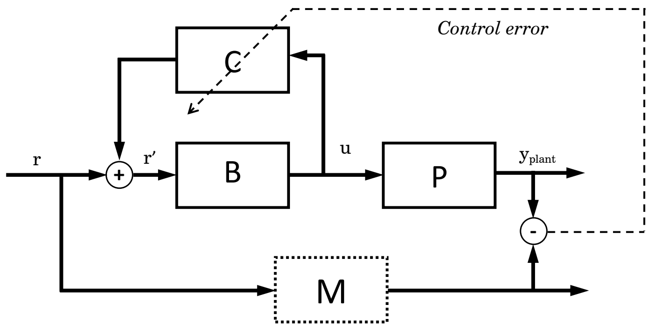


Fig. 1: The Recurrent Decorrelation Control (RDC) architecture used by Lenz et al. [11] to model the Vestibulo-Ocular Reflex. B and C form a recurrent loop, where B is a linear model of the brainstem, and C is a model of the cerebellum realized as a single adaptive filter. P is the plant to be controlled and M is a reference model specifying the desired behavior of the plant.

2 Methods

2.1 Modular RDC Architecture

The Modular-RDC architecture builds on the Recurrent Decorrelation Controller. Here the single cerebellar model is replaced with a set of n parallel modules, each containing a forward and inverse model (Figure 2). Both inverse and forward models are adaptive and take in an efferent copy of motor command. The forward models (F_i) predict the plant state, while the inverse models (C_i) control the plant by modifying the incoming reference signal. In actual fact C_i does not approximate the inverse of the plant, but instead the function $(B^{-1} - P)M$. However, for simplicity C_i is referred to as an inverse model, while the Modular-RDC is referred to as the controller.

The benefit of using multiple modules is that previously learned control can be stored without being overwritten. However, this raises the module selection problem, *i.e.*, how to combine the inverse model outputs to control the plant.

The inverse model outputs are combined in a weighted sum, which is then added to the reference signal. The weights are referred to as the module's responsibility factor λ_i. The responsibility factors are updated at every time-step, and are derived from the forward models' prediction accuracies (see later). It is assumed that modules with good plant prediction have a good controller for the current context. Therefore, the module selection problem can be recast as a plant identification problem.

The feedforward controller (B) presents an opportunity to include any known information about the system, and would usually be designed to control a static approximation of the plant. Here we do not include any information, leaving it as a simple two step delay (z^{-2}). This is done to put all of the control responsibility on the modules, and to highlight its performance in learning good control for a context-switching plant.

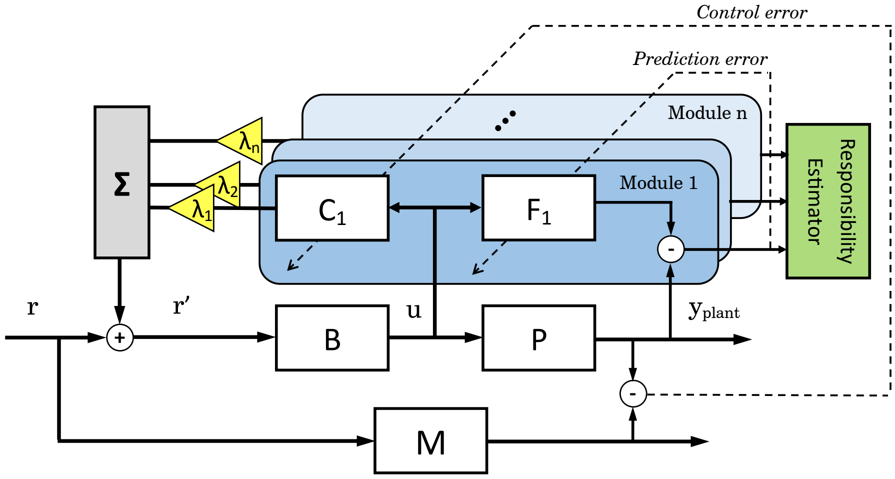


Fig. 2: The proposed Modular-RDC architecture. B is a fixed linear feedforward controller, P is the context-switching plant and M is a reference model with the dynamics desired for the controlled plant. Forward (F_i) and Inverse (C_i) model pairs, together with their responsibility factor constitute a module

The plant to be controlled is of the form:

$$P(s) = \frac{s}{s+a} \tag{1}$$

This simple first-order linear plant was used in Lenz et al. [9] as a model of the oculomotor plant. Different contexts are created by changing the value for parameter a. Most of the results were obtained using three contexts, labeled as A ($a = 10$), B ($a = 40$) and C ($a = 25$). The reference model is defined as:

$$M(s) = \frac{100}{s+100} \tag{2}$$

Forward and Inverse Models Forward and inverse models are implemented using adaptive filters, in accordance with the Fujita model of the microzone (Figure 3). We follow the approach in Wilson et al. [10] and expand the incoming signals through a parallel set of alpha functions G_j . The alpha functions are defined in Equation 3. A large number (50) of alpha functions were used linearly spaced in the range $0.5 - 200$ ms.

$$G_j(s) = \frac{1}{(T_j s+1)^2} \tag{3}$$

Where T_j is the time constant of the j$^{\text{th}}$ alpha function.

The forward and inverse models are trained with the gradient descent rule using prediction errors and the control error respectively (Equations 4 and 5).

In mammalian brains errors become available after a delay of 150-250ms.[1] Although simple to implement, for computational simplicity no feedback delays were included. The learning rate of each module is controlled by a global learning rate $\eta = 0.01$ and the module's responsibility factor λ_i. Linking the module's learning rate to its responsibility factor ensures that modules that are highly involved in controlling the plant improve, while those with little engagement remain unchanged.

$$\Delta w_{ij} = -\eta\lambda_i \, q_j \, (Y_M - Y_P) \tag{4}$$

$$\Delta \phi_{ij} = -\eta\lambda_i \, x_j \, (Y_{F_i} - Y_P) \tag{5}$$

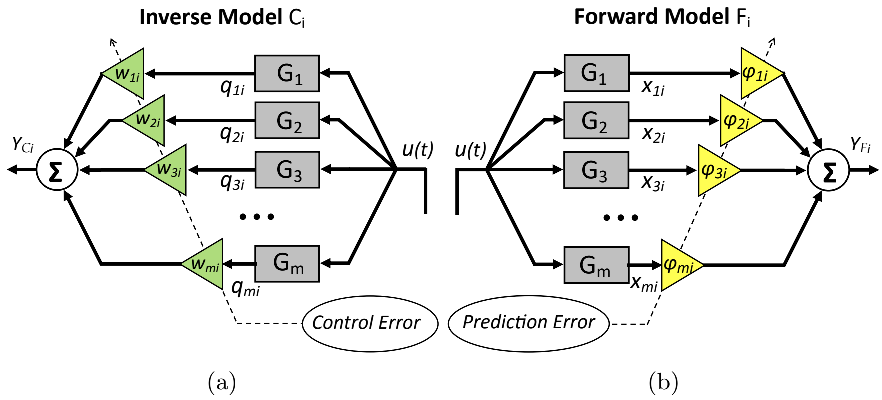


Fig. 3: Adaptive Linear Filter formulations of (a) Inverse and (b) Forward Models

Module Selection As previously stated the module selection problem has been transformed in to the task of plant identification. A responsibility factor can be viewed as the probability that each forward model represents the dynamics of the current plant context. As a set of probabilities, the responsibility factors have the following constraints :

$$1 \geq \lambda_i \geq 0 \tag{6}$$

$$\sum_{i=0}^{n} \lambda_i = 1 \tag{7}$$

In section 3 a new method (**Modified Narendra** method) is compared to three alternative schemes, namely **hard-max** selection, **soft-max** selection used by Wolpert in the MOSAIC architecture[3], and the **original Narendra** method used by Narendra and Balakrishnan [12].

The Original Narendra method selects the controller according to a cost function(Equation 8). The module with the lowest cost function controls the plant. The cost function considers both current and previous errors ϵ_t, with the relative importance of the two elements controlled by parameters α and β. The method has a finite amount of memory T. The following parameter values were found to give good results: $\alpha = 0.5$, $\beta = 0.5$, T = 500 time-steps (0.5 seconds).

The Modified Narendra scheme developed here makes use of the same cost function, but allows for continuous values of λ and the sharing of responsibility across all modules. The responsibility factors are calculated as the inverse of the cost function J normalized across all modules (Equation 9).

$$J_{t,i} = \alpha\epsilon_{t,i}^2 + \beta\sum_{t-T}^{t}\epsilon_{\tau,i}^2 d\tau \tag{8}$$

$$\lambda_i = \frac{1/J_i}{\sum\limits_{k=0}^{n} 1/J_k} \tag{9}$$

The hard-max selection method selects one module with the smallest current prediction error. This method only takes the error of the current time-step into consideration, so is likely to be highly susceptible to noise.

In the MOSAIC architecture, the responsibility factors are calculated using a soft-max function over the current prediction errors (Equation 10) Like the hard-max scheme it can be highly susceptible to disturbance and noise. The manually chosen parameter σ is analogous to the temperature parameter, and determines the level of co-operation and competition between the modules. As $\sigma \to \infty$ all modules share equal responsibility, while $\sigma \to 0$ ensures that the best module receives a responsibility approaching 1. In this work good results were achieved using $\sigma = 0.01$.

$$\lambda_i = \frac{e^{-\frac{|\epsilon_i|^2}{\sigma^2}}}{\sum\limits_{k=1}^{n} e^{-\frac{|\epsilon_k|^2}{\sigma^2}}} \tag{10}$$

2.2 Simulation Set-Up

The simulation was implemented in C++ and was run with a time step of 1 ms. The reference signal consists of colored noise (0.5 - 2.0 Hz) with a distribution in the range of ±1.0 m. Alpha functions, filters, plant contexts and the reference model were implemented using IIR filters. In this work the number of modules is determined off-line, and equals or exceeds the number of plant contexts in a simulation. In future works it is hoped that the number of modules will grow dynamically.

3 Results

3.1 Reduced Transient Errors

The Modular-RDC with Modified Narendra switching was compared to the original RDC controller in a simple scenario with changing contexts. This experiment was developed to test both controllers' abilities to handle context changes, such as those arising from contact with the environment or task switching.

In a training phase lasting 40 seconds, the controllers learned the control for contexts A and B (20 seconds each). For the Modular-RDC controller, the learning was split between two modules, each one specializing in one of the two contexts. The test phase consisted of 20 seconds of controlling context A followed by the final 20 seconds of simulation controlling context B. Both controllers continued to learn throughout the control phase.

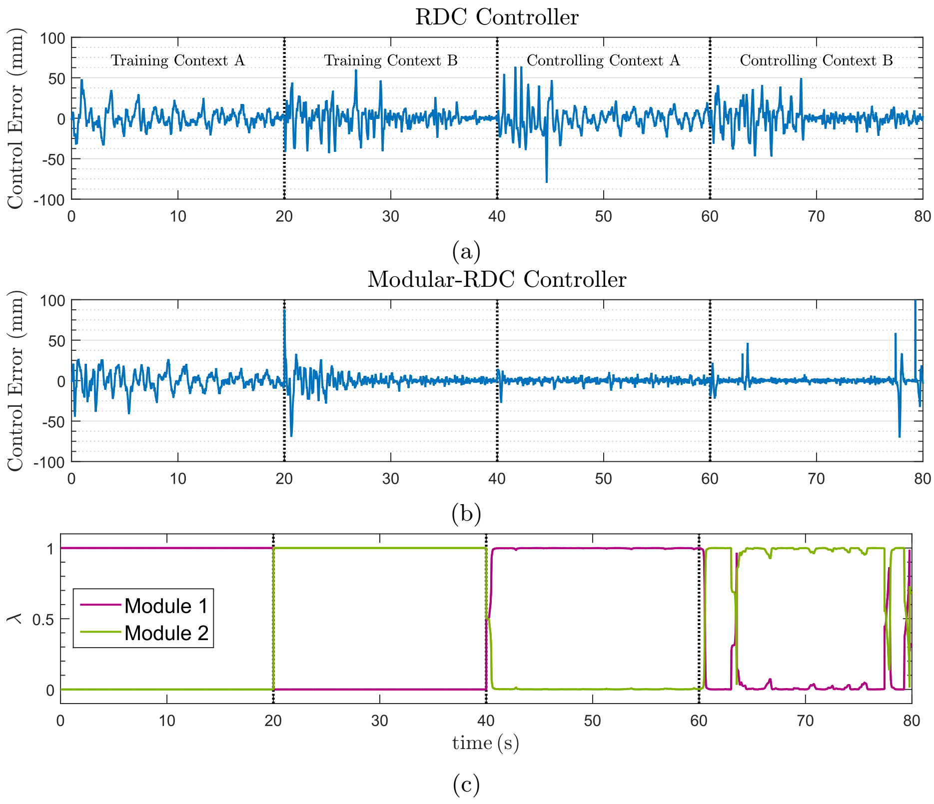


Fig. 4: Comparison of RDC and Modular-RDC controllers during training (A→B) and control (A→B) phases (a) RDC control error (b) Modular-RDC control error (c) Evolution of responsibility factors λ_1 and λ_2

Figure 4c shows the responsibility factors of the Modular-RDC controller throughout the simulation. It can be seen that the proposed controller is able to correctly identify a change in context and select the appropriate module.[1]

Figure 4a shows that the RDC controller error increases after each context change, then decays exponentially as the controller re-learns the control. In Figure 4b the Modular-RDC shows the same behavior during training, *i.e.* when first encountering a novel context. However, in future encounters there is a significantly smaller increase in error. Peak errors resulting from context switching were reduced by 56% and 78% during the test phase, while the total error in this phase saw a reduction of 70.9%. The training phase also saw a 16.3% reduction in average error.

Large errors can be seen at $t = 63\,\mathrm{s}$ and $t = 77\,\mathrm{s}$. These were caused by occasional instability in the forward models. Rapid increases in error could be damaging to equipment and a safety risk. This problem will be addressed in future work.

3.2 Performance of Module Selection Methods

In this section the switching mechanisms described in Section 2.1 are compared. A Modular-RDC controller with two modules was trained to control context A and context B as above. During the test phase, the plant switched between context A, B, and then C, with each context lasting 20 seconds. No further learning occurred during the training phase, as only the controller's ability to identify the current context is tested.

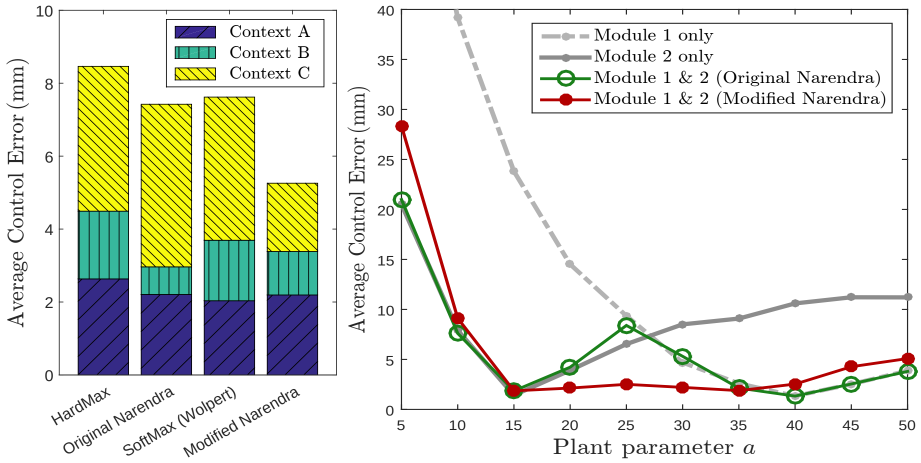
Figure 5a shows the average control error and standard deviation for each of the switching mechanism over 5 runs. It also breaks the average error down to its contributions from controlling contexts A and B (the pre-learned contexts), and the novel context C.

The Modified Narendra method shows significant improvements in the control of both pre-learned and novel contexts. This method is also relatively consistent, with a standard deviation of 8.5% of the average error.

Hard-max selection resulted in an average error of 8.46×10^{-3}. The average error when controlling contexts A and B was 6.74 mm, while for context C this was significantly higher (11.9 mm), 76.9% higher. It can be seen that the switch method performs significantly worse when faced with a novel context compared to a pre-learned one. The original Narendra and Wolpert mechanisms perform moderately better on average than the switch mechanism. However their inconsistency means that they often have similar control errors. Looking at the different contributions to this error reveals that these controllers have a modest improvement in controlling contexts A and B, but are no better in controlling context C.

[1] During the training phase, the responsibility factors are manually set and not determined by the switching mechanism.

9



(a) (b)

Fig. 5: (a) Average control errors resulting from the four module selection methods, showing the relative contributions from each context (b) Control error When Using i) Module 1, ii) Module 2, Module 1 & 2 with two different module selection methods, showing generalization in the Modified Narendra method

3.3 Generalization

The improvements seen in the control of context C, as exhibited in the modified Narendra scheme, suggest that combining controller outputs can lead to generalization of control. Generalization wasn't seen with methods where responsibility is either binary or fluctuates rapidly between modules. To test this hypothesis the original and modified Narendra mechanisms were compared as representatives for the two types of behavior they exhibit.

Figure 5b shows the average control error against plant parameter a for four set-ups of the Modular-RDC controller. The first two plots show the average error when using only one of the two modules. This illustrates that as expected error grows as the difference between the controlled context and the training context increases. Also plotted is the performance of both modules with the *Original Nerendra* and *modified Narendra* selection schemes.

Combining the two modules with modified Narendra switching results in a significant improvement in control for contexts with $10 < a < 40$. At $a = 25$, i.e. mid-way between contexts A and B the average error was reduced from 12.60 mm to 3.76 mm, a reduction of 70.1%. This shows that the Modular-RDC can generalize between pre-learned controllers in parameter space. However, results for $a < 10$ and $a > 40$ suggest that this generalization does not extend outside of this range. It can be seen that the *original Narendra* results in no generalization, and its control is only as good as the better of the two controllers. It must be noted however that this is still a significant improvement on using a single controller, as is the case with RDC control.

4 Discussion

In this work we have applied the simple yet effective concept of distributed control to the RDC architecture. The resulting control architecture, Modular-RDC has been shown to reduce transient errors resulting from a sudden change in plant dynamics. This allows for the Recurrent Decorrelation Control paradigm to be applied to a new class of system, i.e. rapidly time-varying plants. Moreover, it was shown that any of a number of module selection strategies were able to correctly identify a change in context. The best module selection scheme (Modified Narendra) resulted in generalized control of plant dynamics in between pre-learned contexts.

The Modular-RDC is similar to the architectures developed by Wolpert et al. and Narendra and Balakrishnan. All three systems convert the module selection problem into a plant identification problem through the use of forward models. A direct comparison between the three systems would be outside the scope of this study, which is to improve and expand the RDC architecture. However, here some prominent differences are briefly addressed.

Narendra's controller uses one controller at a time to drive the plant. The architecture also relies primarily on fixed controllers. The combination of few adaptive elements and no sharing of responsibility means that many controllers are needed so as to cover the plant's parameter space. The most significant difference between Modular-RDC and MOSAIC is that they are founded on contrary bio-inspired control architectures, RDC and Feedback Error Learning (FEL) respectively. These two controllers differ fundamentally in that RDC is a feedforward controller, while FEL uses feedback. Also, they present different solutions to the distal teacher problem for adaptive controllers. The RDC controller allows for faster feedforward control that is robust to large feedback delays.

In this work it was shown that combining the outputs of inverse models leads to generalization of control to some novel contexts. It suggests that a small number of modules can effectively control many contexts. Furthermore, clinical evidence suggests that generalization can be extended beyond the range seen in Section 3.3.[14] Generalized control could result in greater computational efficiency and good control of novel contexts.

This work has demonstrated significantly improved control for a simple single-DoF linear system. In future studies we intend to look at the control of more complex nonlinear and multi-DoF systems with sensory delays implemented. It is also hoped that the Modular-RDC controller can act as a biologically plausible model of cerebellar control.

The Modular-RDC formulated here presents opportunities for improvement. For example, re-evaluating the module selection problem as a dynamic unsupervised/reinforcement learning problem can provide more scientific rigor and should improve module selection. Also, allowing the controller to add modules during operation would allow for the system to adapt on-line to meet real-world challenges. Finally, methods for ascertain an optimal number and range of α functions could be developed.

5 Conclusion

This work has created a distributed controller around the adaptive Recurrent Decorrelation Control (RDC) architecture to improve control of a context-switching plant. The result is the Modular-RDC controller. During simulations with a context switching 1^{st} order plant, the Modular-RDC controller was shown to result in a 70.9% reduction in control error during the control phase. The controller achieves this by correctly identifying a change in dynamics and selecting the appropriate module to control the plant.

The Modular-RDC controller has a degree of generalized control, as it was able to effectively control a novel context with plant parameter a between two pre-learned contexts. The combined effort of both modules resulted in a significantly lower error than either module could achieve on its own. This generalization was shown to be limited to contexts with plant parameter a within the range of pre-learned contexts, and did not extend beyond this range under the current module selection scheme.

Bibliography

[1] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

[2] Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7):1317–1329, 1998. doi: 10.1016/S0893-6080(98)00066-5.

[3] Daniel M Wolpert, R Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9):338–347, 1998.

[4] Hiroshi Imamizu, Tomoe Kuroda, Satoru Miyauchi, Toshinori Yoshioka, and Mitsuo Kawato. Modular organization of internal models of tools in the human cerebellum. *Proceedings of the National Academy of Sciences*, 100(9):5461–5466, 2003.

[5] Masao Ito. Cerebellar circuitry as a neuronal machine. *Progress in neurobiology*, 78(3):272–303, 2006.

[6] Masao Ito. Historical review of the significance of the cerebellum and the role of purkinje cells in motor learning. *Annals of the New York Academy of Sciences*, 978(1):273–288, 2002.

[7] M Fujita. Adaptive filter model of the cerebellum. *Biological cybernetics*, 45(3):195–206, 1982.

[8] John Porrill, Paul Dean, and James V Stone. Recurrent cerebellar architecture solves the motor-error problem. *Proceedings of the Royal Society of London-B*, 271(1541):789–796, 2004.

[9] Alex Lenz, T Balakrishnan, Anthony G Pipe, and Chris Melhuish. An adaptive gaze stabilization controller inspired by the vestibulo-ocular reflex. *Bioinspiration & biomimetics*, 3(3):035001, 2008.

[10] Emma D Wilson, Tareq Assaf, Martin J Pearson, Jonathan M Rossiter, Sean R Anderson, and John Porrill. Bioinspired adaptive control for artificial muscles. In *Conference on Biomimetic and Biohybrid Systems*, pages 311–322. Springer, 2013.

[11] Alexander Lenz, Sean R Anderson, Anthony G Pipe, Chris Melhuish, Paul Dean, and John Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1420–1433, 2009.

[12] Kumpati S Narendra and Jeyendran Balakrishnan. Improving transient response of adaptive control systems using multiple models and switching. *IEEE Transactions on Automatic Control*, 39(9):1861–1866, 1994.

[13] Masahiko Haruno, Daniel M Wolpert, and Mitsuo Kawato. Mosaic model for sensorimotor learning and control. *Neural computation*, 13(10):2201–2220, 2001.

[14] Paul R Davidson and Daniel M Wolpert. Internal models underlying grasp can be additively combined. *Experimental Brain Research*, 155(3):334–340, 2004.