A Developmental Organization for Robot Behavior

Roderic A. Grupen
Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts Amherst
grupen@cs.umass.edu

Abstract

This paper focuses on exploring how learning and development can be *structured* in synthetic (robot) systems. We present a developmental *assembler* for constructing reusable and temporally extended actions in a sequence. The discussion adopts the traditions of dynamic pattern theory in which behavior is an artifact of coupled dynamical systems with a number of controllable degrees of freedom. In our model, the events that delineate control decisions are derived from the pattern of (dis)equilibria on a working subset of sensorimotor policies. We show how this architecture can be used to accomplish sequential knowledge gathering and representation tasks and provide examples of the kind of developmental milestones that this approach has already produced in our lab.

1. Introduction

Human infants display a tremendous assortment of time-varying structure in their physiological and neurological responses to the world. We propose that kinematic, dynamic, and neurological properties of a developing infant are exploited to simplify and structure learning in the context of an on-going interaction with the world. Developmental processes construct increasingly complex mental representations from a sequence of tractable incremental learning tasks. Our goal is to provide computational mechanisms for modeling these aspects of developmental in order to program complex robot systems.

By Piaget's account, the sensorimotor stage in human infants lasts roughly twenty four months (Piaget, 1952, Piaget, 1954). In the first four months, reflexive responses begin to organize into coherent motor strategies, sensory modalities are coordinated and attentional mechanisms begin to emerge. From four to six months, primary circular reactions are practiced until the infant finds it possible to prolong certain kinds of interactions. Between six and eighteen months, these primary circular reactions lead to behavioral models of the world that apply to "classes" of interactions and depend less on specific instances. This paper explores the possibility that a commitment to such *figurative schemata* explains some qualitative aspects of developmental processes and leads to the efficient acquisition of abstract, hierarchical control knowledge that can be used to similar advantage in the organization of robot behavior.

Two central hypotheses underlie our research:

- rich classes of behavior can be expressed in terms of relatively few, schematic structures that are grounded in physical activity that we call *physical schemata*, and
- physical schemata are the sensorimotor foundation for re-usable control knowledge.

The idea of physical schemata is not new in the behavioral sciences (Johnson, 1987, Lakoff, 1984, Lakoff and Johnson, 1980, Mandler, 1988, Mandler, 1992, Gibbs and Colston, 1995), but we are proposing computational mechanisms whereby a robot can acquire hierarchies of physical schemata, propose generalizations of such schema, and test these propositions. We suggest that cumulative cognitive models both define and explain a robot's relationship to multiple goals in a non-stationary environment.

One implication of this hypothesis is that once stable interactions with the world are discovered and captured as physical schemata, an agent may predict other instances of this type of interaction. This is one form of *metaphorical extension* (Johnson, 1987), whereby physical schemata are extended to other physical examples of that phenomena. We will introduce the notion of symmetries in behavioral descriptions that support the rapid adaptation of a schema to specific run-time contexts. Homomorphisms of physical schema provide a means of learning (or being taught) under what conditions one may explore using parameterized schema rather than learning from scratch. Thus, it can help us avoid brute force, unstructured stochastic search in related contexts and perhaps as a consequence, use development in part to defeat the curse of dimensionality.

2. Structure for Learning and Development

An infant is an integrated, adaptive system - he/she is a complex and flexible "plant" embedded in a fluid and open environment. As a control problem, this is truly daunting. But, we advocate a relatively optimistic position - that traditions in robotics, control theory, AI, and learning are adequate computational accounts of some critical aspects of developing human infants. An important lesson to learn from development is that embedded sensorimotor processes can compromise expressive power temporarily to reduce complexity. Managing this tradeoff effectively can lead to computational tractability in the short term and growth toward optimal behavior in the long term.

2.1 Kinematic, Dynamic, and Maturational Structure

Traditions in machine design have often applied tools for predicting kinematic function and dynamic effects in the motion of an articulated structure. Roboticists, in turn, have often used these tools to fashion mechanisms with appropriate kinematic and dynamic properties. However, our ability to address open tasks in unstructured environments remains *ad hoc* and limited to the ingenuity and foresight of the designer. For instance, Salisbury (Salisbury, 1982) designed robot hands so as to be kinematically isotropic in the region where individual fingers may collectively manipulate an object. In an open domain, when object geometry and task are unspecified, this heuristic places all three fingers near isotropic configurations in the neighborhood of commonly used hand postures.

The value of intrinsic dynamics in robot design is perhaps most convincingly demonstrated in the development of passive bipedal walkers. Researchers have built passive mechanisms based on human morphology that rely almost entirely on the passive dynamics of the mechanism (McGeer, 1992, McGeer, 1993).

In humans, kinematic properties of the skeleton and the dynamics of the musculature strongly influence the way in which development proceeds. For instance, Bernstein seminal work observed periods of kinematic engagement followed by the formation of strategies to exploit non-muscular (inertial) forces during movement(Bernstein, 1967).

In addition to a "kinematic-then-dynamic" strategy, the dimensionality of the initial kinematic problem can be addressed in a staged manner as well. Berthier, Clifton, McCall, and Robin (Berthier et al., 1999) conducted longitudinal studies of infants (6 - 30 weeks) during the onset of visually- and acoustically-guided reaching tasks. Initially, reaching movements appear to be focused primarily in the shoulder and torso. Large proximal degrees-of-freedom are engaged first while the intrinsic muscles of the forearm and hand stiffened via co-contraction. Researchers (Kuypers, 1981) propose that patterns of maturation in the corticospinal tract are responsible for this kind of structured learning.

Bril and Brenière (Bril and Brenière, 1992) make similar observations in the development of bipedal gaits. In the first 5 months, infants integrate postural constraints into gait generation. A coordinated gait can be elicited immediately after birth in the form of the gait reflex (Thelen and Smith, 1994). However, the reflex is segmental and not integrated with other skills required for upright locomotion. During an important developmental period, a conjunction of influences inhibits or suppresses the gait reflex in order to permit the stiff-legged (or distal co-contracted) infant to focus on postural stability. These walking strategies are focused primarily in the hip, with relatively wide stance. Once muscle strength is adequate and policies for balance and equilibrium are established, then distal degrees of freedom are enlisted. A co-contraction heuristic appears to be a common motor policy during the acquisition of new motor skills even in adults (notice how adults approach skiing or ice skating as novices).

2.2 Reflexes and Composability

The CNS is organized, not in terms of anatomic segments, but according to *movement patterns*(Aronson, 1981). The basic form of packaged movement pattern is the reflex. A reflex can reside at many levels of the central and peripheral nervous system ranging from involuntary responses to trigger stimuli, i.e. muscle stretch, the so-called simple segmental reflexes mediated in the spinal cord and brain stem, postural reflexes mediated in the cerebellum, and even cortically mediated visual reflex. These processes contribute to the organization of behavior at the most basic levels - they constitute a kind of sensorimotor instruction set for the developing organism. The so-called developmental reflexes, whose onset and persistence have been meticulously documented, serve ontogenetic developmental goals and are not elicited in normal adults. In addition to providing basic sensorimotor function, it is generally understood that they exercise the musculature and serve to increase the exposure of learning and developmental processes to conditions underlying important developmental milestones.

Composition of such sensorimotor systems gives rise to more comprehensive behavior, For example, in (Mussa-Ivaldi et al., 1991) it is observed that individual force fields are superimposed in the frog's leg/spine that yield continuously controllable leg po-

sition from a discrete set of composable force fields. Moreover, in humans it is generally understood that certain motor patterns repeat in a regular pattern. Some, like walking, swimming, or flying, are the result of specialized ensembles of cells in the CNS called Central Pattern Generators (CPGs). Wolff (Wolff, 1991) suggests methods for composing oscillators in order to address novel initial conditions and contexts. For example, infant breathing rhythms vary between 30-40 cycles per minute in a rough balanced inhale/exhale episodes. When an infant is startled a repeatable change in the amplitude and frequency can be observed that disappears 3-4 breathing cycles after the stimulation disappears. Biological rhythms can thus be perturbed by superimposing other such rhythms and can return spontaneously from neighboring trajectories to the same "attractor basin" when the perturbation is removed (Wolff, 1991). Similar ideas underlie recent work by Williamson (Williamson, 1999) who demonstrated oscillators that couple across kinematic structures.

3. A Developmental Assembler

The epigenetic developmental theory proposes that primitive reflexes, expressed as neuro-anatomical structures, are the basis building blocks of behavior. In this model, complex behavior is constructed from combinations of primitive reflexes in response to reinforcement. Ontogenetic developmental theory suggests that coordinated behavior appears and subsequently disappears in order to serve some vital developmental function. After these developmental roles are fulfilled, the neural substrate of the behavior is eliminated as well - it does not grow or otherwise change into a mature form of the reflex. This implies that some types of behavior do not have an antecedent in a reflexive repertoire.

We contend that reflexes serve as an epigenetic computational basis and that some are short-lived and serve an ontogenetic knowledge formation role. For example, the stepping reflex is likely the antecedent of walking, but no such simple reflexive precursor has been identified for reaching tasks. The latter being a discrete task, requiring multiple coordinated reflexes (Asymmetric Tonic Neck Reflex (ATNR), palmar grasp reflex, distal-curl reflex, Moro (clasp) reflex, startle, etc.). To understand developmental processes, therefore, we need to understand how knowledge and structure interact over time to acquire skill in a manner that is robust with respect to variation in the world. Furthermore, to implement developmental processes there must exist a generative basis for behavior that coordinates simple reflex to serve both epigenetic and ontogenetic developmental goals, there must be an internal representation for control knowledge, and a principled framework for reusing that knowledge in novel circumstances.

Figure 1 is a sketch of a computational framework that addresses both learning and development and that incorporates some of the principles of structure discussed earlier. One dimension of development is viewed as a scheduling problem in which a strategy for engaging sensor, motor and computational resources is sought to satisfy a task specification incrementally. Each stage of this process is characterized by developmental parameters; the tasks, the participating control objectives, the sensor and effector resources allocated, and axioms that define legal combinations of behavior. The overall objective is to progress through a sequence of such designs to assemble new behavior, which in turn, becomes a precursor for successive stages of development.

The rest of Section 3. will introduce the main parts of the architecture in Figure 1. After this introduction, we will review a developmental sequence realized on platforms in our lab and speculate about what's next.

3.1 Action

Our framework is designed to learn a hierarchy of actions by composing more primitive actions. The **Control-Basis** (Huber et al., 1996, Coelho and Grupen, 1997) in Figure 1 is designed to provide a combinatoric basis for control that supports the representation of declarative and procedural control knowledge. The most primitive actions are closed-loop control processes constructed by combining an artificial potential, $\phi \in \Phi$, with a subset of the available "sensors," $s \in \Omega_s$, and "effectors," $e \in \Omega_e$. The effect of action plays out over an extended period of time as a pattern of system control inputs act to descend the potential function using the available resources. The "control basis" terminology highlights the fact that there are many possible sensor/effector assignments for each type of artificial potential.

Modeling primitive actions as controllers has several important related consequences:

- actions are asymptotically stable and tend toward fixed points that represent locally optimal conditions with respect to the objective function;
- controllers suppress local perturbations by virtue of their closed-loop structure;
- the dynamics of the controlled system provides a variety of useful discrete abstractions of the underlying continuous state space; and
- time is metered by discrete observable events in the transient response of the controlled system rather than by an arbitrary continuous clock.

Since dynamics reveal events that depend on the operational context, hierarchies of *actors* are likewise

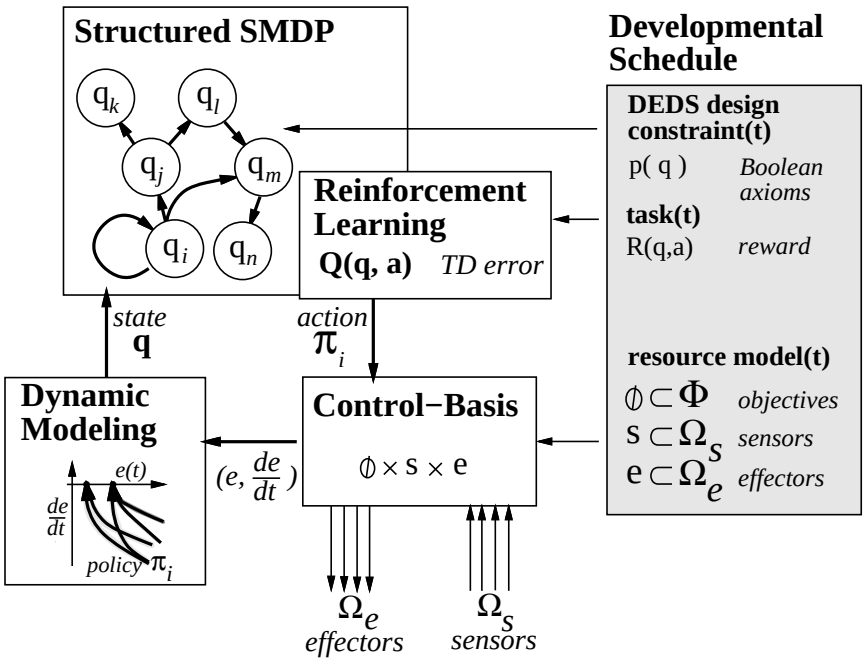

Figure 1: Native Structure, Learning, and Behavior in an Integrated Developmental Assembler.

hierarchies of special purpose *observers*.

Artificial potentials also share critical properties with *value functions* employed by the machine learning community to make optimal control decisions. In particular, value functions preserve all the properties of *admissible* controllers, leading to naturally hierarchical organizations of controllers defined in terms of other controllers. These hierarchies express objectives at many temporal scales so that an agent can adjust its level of discourse about activities by ascending and descending the hierarchy. Moreover, when a robot receives procedural instruction from another agent (human), it can estimate the *purpose* of these instructions in light of its own hierarchy of objectives, as well.

This perspective is inspired by some recent trends in robotics and the relationship of these ideas to epigenetic theories of development. In 1989, Koditschek *et. al.* argued that robot designers should focus on "finding controllers" with inherent dynamical properties that produce valuable artifacts in the world rather than computing the artifacts directly (Rimon and Koditschek, 1989). Assertions about the stability of the coupled system have been used to form the state space for such systems as in the attractor landscape proposed by Huber *et. al.* (Huber and Grupen, 1997) or the limit cycles proposed by Schaal *et. al.* (Schaal and Sternad, 1998).

3.2 State and System Modeling

Consider the case when a dexterous robot must grasp an object. The dominant traditions in grasp planning require that the complete geometry of the object be determined *a priori* so that a planner can enumerate all possible grasps and sort them into an order that reflects their relative quality. At this point, the robot will execute the solution at the top of the list. In contrast, the control basis approach advocates searching for control configurations that produce high quality equilibria. This is an on-line process that does not require a complete object geometry.

The **Dynamic Modeling** component of Figure 1 is responsible for modeling the signature dynamics of the controlled process in many operational contexts. For example, Figure 2 plots the potential (a candidate Lyapunov function) against the time rate of change of the potential for a grasp controller that descends toward *wrench closure* grasp configurations in a variety of grasp contexts (Coelho and Grupen, 1997). The controller has multiple equilibria because there are typically many objects that can be grasped and many different grasp solutions for any one object. When policy, π_i, is engaged, the pattern of membership in these empirical models changes over time in a manner that identifies the current control context. This perspective has roots in methods like Hidden Markov Modeling (HMM) where categories can be recognized by parsing a sequence of events according to a transition model. Likewise, Takens's theorem describes how patterns in the behavior of nonlinear dynamical systems are related to missing or hidden state variables. In our architecture, closed-loop controllers produce

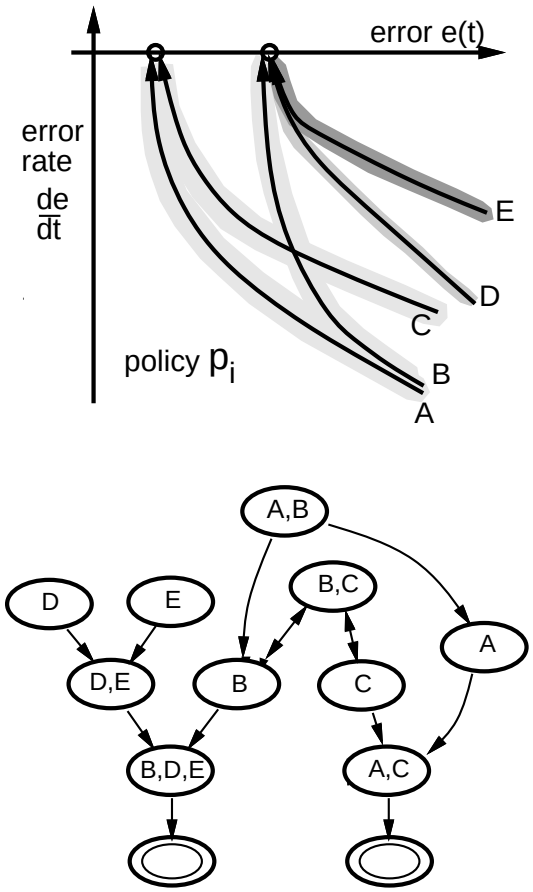

Figure 2: The pattern of membership in governing dynamic models serves to identify a discrete state for the control policy.

mechanical artifacts *and* an event stream that distinguishes certain control contexts.

As discussed previously, controllers are distinguished by their respective sensory and motor resource allocations. The goal for controller ϕ_i is to achieve an equilibrium state where its value function is at an extremum. We want to determine whether an estimate of the run-time context can be extracted from the transient response of ϕ_i. We define a predicate vector $\vec{q}_i$ that describes the status of ϕ_i by identifying the set of empirical models M_{ij} that are consistent with a series of run-time observations. For instance, an element of $\vec{q}_i$ can represent convergence - asserting that ϕ_i has reached equilibrium. Under these circumstances, a particular stance of a walking machine is stable, or the robot arm has achieved a reference configuration. Other elements of $\vec{q}_i$ assert that run-time observations match other M_{ij}. In (Coelho, 2001), we found that a small set of such models can be used to recover a wide variety of interesting control contexts.

In general, an agent's predicate state $\vec{q}$ reflects the current status of several active controllers. We denote by $p(q_{ij}|\phi_i(\vec{q}))$ the probability that model M_{ij} explains the observed time history when control ϕ_i is engaged in state $\vec{q}$. The *system identification* task is to learn $p(q_{ij}|\phi_i(\vec{q}))$ for all predicate states $\vec{q}$. These probabilities constitute a *physical model* of the world couched in the agent's interaction with it. The resulting transition model describes the dynamics of information gathering as well as grasp formation under policy π_i.

3.3 Developmental Schedule

The set of primitive actions that may be expressed, $\Phi \times 2^{\Omega_s} \times 2^{\Omega_e}$, is quite large. This is good from the perspective of expressive power but bad from the standpoint of computational complexity and policy formation. So while the system may be capable of associating novel combinations of features in the sensor data with unorthodoxed combinations of effectors to accomplish a specific stimulus-response process, it is not clear that these unlikely actions will ever be explored by purely randomized search. Therefore, several aspects of developmental structure have been implemented in the Developmental Schedule component of Figure 1 to bias exploration toward computationally tractable subsets of the action and state sets in order to accumulate critical control knowledge sequentially.

The **resource model** expresses constraints on the sensors, effectors, and potential functions/policies that may be considered when generating actions. It is this kind of supervision that underlies the proximal to distal motor engagement during the onset of reaching movements and bipedalism (Section 2.).

The sequence of **tasks** that the robot addresses can clearly influence the performance of the developing system. We will present and example in which the right task sequence has a significant impact on learning performance and the asymptotic quality of the policies acquired in Section 4.3.

The **Discrete Event Dynamic Systems (DEDS)** specification provides the finest grain control over the combinatorics of exploration. It constrains the range of interactions permitted with the environment to those that:

- satisfy real-time computing constraints;
- guarantee safety specifications; and
- are consistent with kinematic and dynamic limitations.

In this formalism (Özveren and Willsky, 1990, Ramadge and Wonham, 1989, Sobh et al., 1994), the state of the underlying system is assumed to evolve with the occurrence of a set of discrete events, some subset of which are controllable. There many tools for analyzing and interacting with such control processes. One may prove, for instance, that certain states can't occur. These tools also provide a means of investigating the role of constraints as "bootstraps" for a learning system. Such a

mechanism influences the occurrence of controllable events such that no prohibited or uncontrollable event can violate functional constraints on the system. A complete supervisor takes the form of a nondeterministic finite state automaton in which states are functional assertions about patterns of membership in the empirical dynamic models that must be either preserved or excluded and transitions represent possible concurrent control situations.

Logical conditions on the predicate vector, influence the range of control options that the system may explore. In a similar fashion, this DEDS supervisor allows the introduction of additional domain knowledge and preferences into the control architecture. Doing so can dramatically reduce the number of control alternatives considered in the learning phase and can be used to accelerate learning or as a shaping mechanism.

3.4 Reinforcement Learning

Dynamic programming-based Reinforcement Learning (RL) (Barto et al., 1993) is a natural paradigm for programming these systems since RL does not require external supervision and encodes policies as *sequences* of concurrent control situations with associated rewards. In general, these rewards can be rare, occurring infrequently or after extended sequences of actions. In the framework presented here, RL is used to solve the temporal credit assignment problem for an optimal policy with respect to a given reinforcer.

One of the major drawbacks of reinforcement learning methods is the large number of trials they require in order to first find and then improve a given policy. A second problem in exploration-based learning techniques is that they have to take random actions in order to discover better control policies. This can lead to catastrophic failures which are not acceptable for an autonomous system. The mechanisms outlined in Section 3.3 are designed to address these shortcomings and lead to high performance learning systems.

As useful policies for important tasks are constructed, they are incorporated into the control basis. Therefore, subsequent policies may incorporate *temporally extended* actions if there is added value to reasoning about the problem at these time scales. This approach is formalized in the Semi-Markov Decision Processes (SMDPs) as a recursive framework for hierarchical control that leads potentially to large expressive power depending on the sensors, motors, the native value functions, and the developmental sequence applied to the agent.

4. Developmental Milestones

In (Fiorentino, 1981), a coarse description of the developmental process during the first year of an infant's life is presented. The discussion focuses on tasks and reflexes that facilitate motor control and perhaps the acquisition of control knowledge. A sequence of postural stability tasks are identified that start with the infant acquiring the ability to control its head. In this kind of task, information is assumed to be heavily weighted toward vestibular, proprioceptive, and (later) vision organs. Figure 3 illustrates one such early sequence in which a child learns to raise its head off the floor (like a mast) into a configuration where the axis of the head is aligned with gravity. This occurs during a period of learning about and strengthening muscles in the neck and the back that are as yet weak in extension. The infant uses optical- and labyrinthine-righting reflexes that are not present at birth but develop instead over the first few weeks. These mechanisms, from the prone position, interact with the symmetric tonic neck reflex to develop a guadrupedal position. A purely proprioceptive reflex called the "body-on-head" reflex helps to rotate the trunk in response to a head angle. The infant thus acquires policies for rotating the trunk and head about the body axis to pan the head and eyes. All of this behavior leads toward stabilizing the infant in sitting and later standing postures. The author describes a set of reflexes, their onset and persistence, that when elicited build range of motion, strength, and compliance or that cause important interactions (hand-eye interactions, for instance).

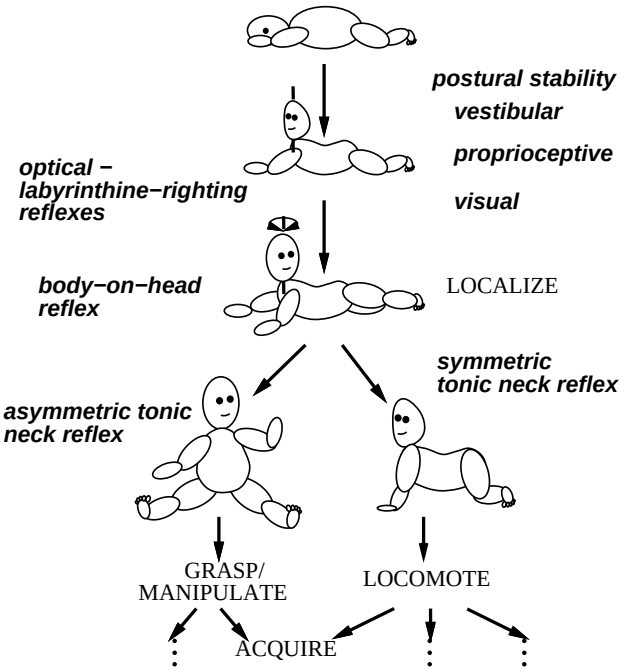

Figure 3: Superimposing a developmental sequence observed in human infants with some of the Schemata developed on robotic platforms.

With the foundations of the previous sections, we will conclude by extrapolating from results generated over the past decade in the Laboratory for Perceptual Robotics at UMass on several experimental platforms to speculate about a staged programming protocol for dexterous robots. The behavior discussed is superimposed in capitals on Figure 3 in order to emphasize the sequential development of sensorimotor programming directed by tasks, resources, and pre-requisite control knowledge.

4.1 LOCALIZE *Schemata*

Two close relatives in the control basis were configured that employ the same artificial potential and effector resources. The only difference is the source of the position reference. In Figure 4, SACCADE, ϕ_s, accepts a reference heading in space and directs the sensor's field-of-view to that heading. FOVEATE, ϕ_f, is similar except that it accepts heading references determined by the sensor's signal. The SACCADE-FOVEATE schema has been applied to scanning infrared sensors, distributed sensor arrays, acoustic sensors, binocular heads, and panoramic visual sensors. The state in the nodes of Figure 4 is just the

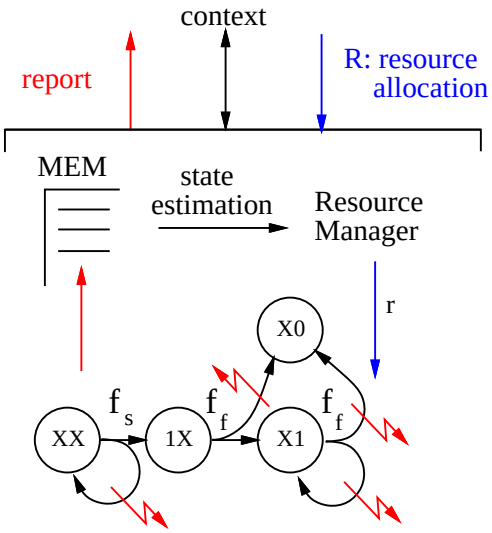

Figure 4: SACCADE-FOVEATE - A Physical Schema for Detecting and Tracking a Feature.

convergence status of ϕ_s and ϕ_f. That is, if ϕ_s is converged and ϕ_f is not, then the state of the SACCADE-FOVEATE schema is 10. An X in the state representation represents a "don't care" or "don't know" condition.

The SACCADE-FOVEATE schema begins by directing a sensor $r \in R$ to saccade to an interesting region of space. A transition from state XX back to state XX is presumably an error in the motor component for the sensor. If the sensor achieves state $1X$, then the closed-loop process ϕ_f is engaged whose goal it is to bring the centroid of the response of a local feature detector to the center or fovea of sensor r's image plane. A transition from $1X$ to $X0$ via controller ϕ_f can signify that the expected target feature is not present. If the target feature is detected and foveated, then the sensor achieves state $X1$ where the feature is actively tracked. As long as the actions of the foveation controller preserve this state, a heading to the feature is reported. If a transition from $X1$ to $X0$ is observed under action ϕ_f, then a target feature may have eluded the sensor. This rich semantic description of the behavior is derived exclusively from the transition dynamics in the SACCADE-FOVEATE schema.

When at least two instances of the SACCADE-FOVEATE schema are simultaneously in state $X1$, and they are driven by features derived from the same subject, then there is sufficient information for triangulating the subject. Figure 5 illustrates a schema that manages a multiple SACCADE-FOVEATE schemata hierarchically. The LOCALIZE schema re-

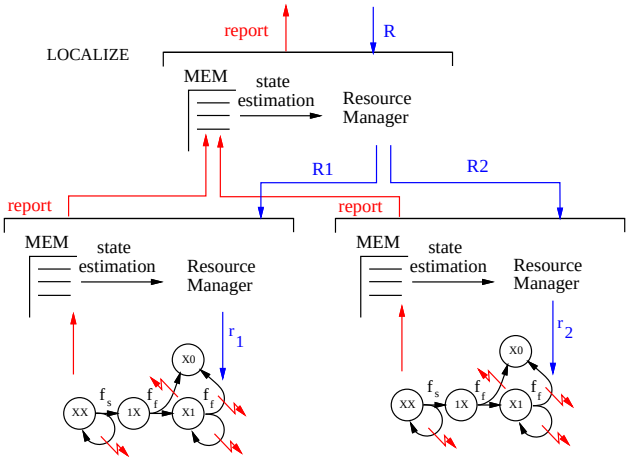

Figure 5: LOCALIZE - A Physical Schema for localizing features in space.

ceives event streams generated by multiple subordinate SACCADE-FOVEATE schemata under its management and produces a report regarding the location of the subject. If the subject is human, then they may at times be moving, stationary, speaking or quiet, possessing biometric features that are robust over time and some that change (like clothing). The LOCALIZE schema must configure a process that is adequate for this purpose. Each unique resource allocation $r_1, r_2 \in R$ produces hypotheses of varying quality depending on the context of the localization query.

We anticipate that the LOCALIZE resource manager must learn when and how to pass the tracking task onto other sensors. In fact, the hand-off might exploit the resource pool by instantiating redundant tracking modes and using their feedback to refine

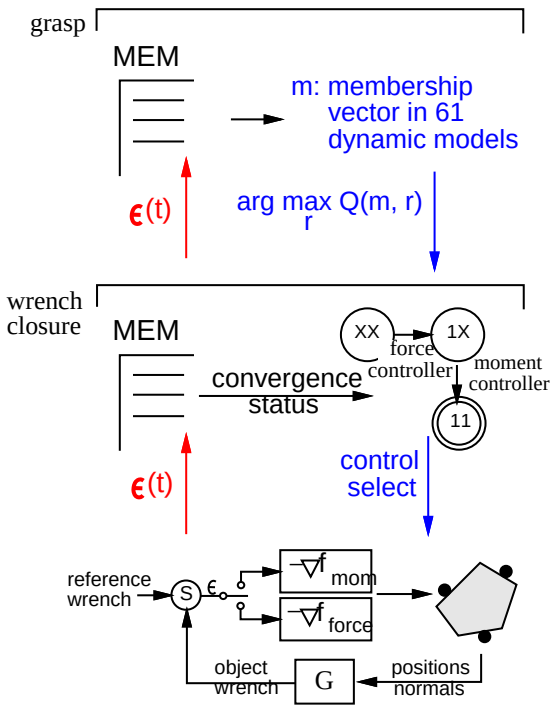

Figure 6: A hierarchical controller developed in stages for grasping unknown objects from the set of convex polytopes.

the control context. This context can include visual, acoustic, and thermal signatures of the subject, but it can also entail, perhaps, partially "recognizing" the subject. We're asking alot of this resource manager - but we're asking these hard questions in a very specific behavioral context.

4.2 GRASP *Schemata*

In (Coelho and Grupen, 1997), we proposed a closed loop grasp controller that generates fixed points in contact configuration space based solely on tactile feedback. The C^* controller is actually a combination of two artificial potentials derived from contact positions and normals and assuming uniform unit magnitude contact forces. The first potential measures a quadratic force residual and the second measures a quadratic moment residual. We showed that a descent on the force residual surface to convergence followed by descent on the moment residual surface produces optimal wrench closure grasps on regular, convex geometries. This policy is captured in the transition diagram in the middle of Figure 6. In subsequent work (Grupen et al., 1992, Coelho, 2001), the "regular/convex" grasp policy was extended hierarchically to account for varying numbers of contact resources and to tolerate bounded irregularity within the class of convex objects. For example, given 3 fingers and the performance metric employed, the optimal number of contacts for grasping a triangle is 3, for a rectangle 2 contacts are optimal, and for the cylinder, both 2 and 3 contacts are optimal. Moreover, grasp configurations that are evolving toward suboptimal equilibria can be shunted toward optimal fixed points by descending alternate potential fields.

We presented a variety of irregular triangular, cylindrical, and rectangular prisms to the system. A total of 61 models (like those illustrated in Figure 2) of the behavior of C^* were generated for these objects. The pattern of membership in these models constitute the state of a Markov Decision Process (MDP). A control decision to reallocate resources, r, is considered whenever membership in these models changes during grasp formation. There are four distinct such parameterizations involving sets of fingers - [1 2 3], [1 2], [1 3], and, [2 3].

One hundred trials were executed for each of three object types within the convex class (a rectangle, a cylinder, and a triangular prism). In each trial, an unknown object was presented and one of the 4 grasp controllers was executed until convergence at which point the grasp configuration was evaluated. Figure 7(a) presents the distribution of grasp scores obtained. Then, a reinforcement learning algorithm was used to construct a policy for allocating contact resources. Figure 7(b) shows that the mean grasp performance improves and the variance

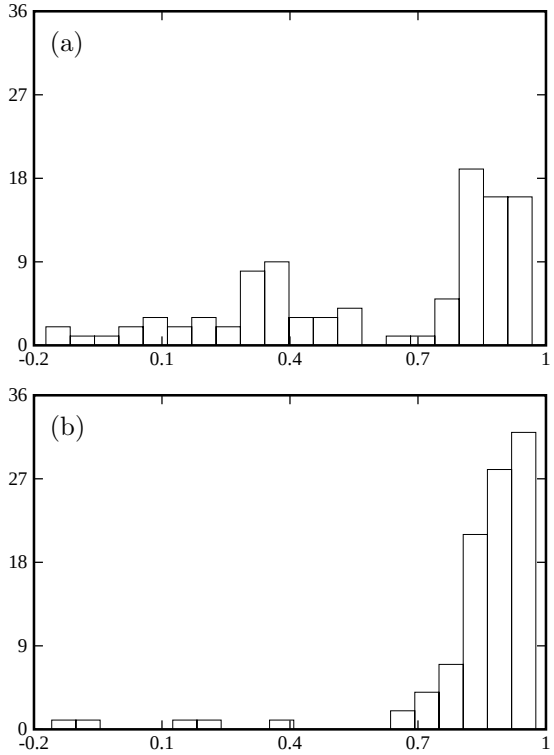

Figure 7: Distribution of grasp scores, over 100 trials; (a) baseline performance using native controllers, (b) impact of policy for resource allocation based on patterns of membership in the empirical models of error dynamics.

in performance decreases as result of learning. In the process of grasping, often the type of object was determined, although this was not the goal of the control process. Similar results were observed when we examined performance on each object. It should be possible to generalize this approach to non-convex objects as well, although we have not yet attempted to do so.

4.3 LOCOMOTE *Schemata*

Multi-legged locomotion platforms present similar challenges to those of multi-fingered robot hands when it comes to coordinated motion planning. Coordinated limb movements must serve both stability and mobility concerns by sequencing the movement of several independent kinematic chains. There are active communities considering gait synthesis for walking platforms and now for manipulation as well, but so far there has not emerged a unified framework for solving these problems.

Our walking platform, "Thing," is a small, twelve degree of freedom, four legged walking robot built to explore whether ideas designed for quasistatic finger gaits could be applied to quasistatic locomotion gaits. Initially, it was given three types of motor actions $\phi \in \Phi$, in the control basis; force, position, and kinematic conditioning controllers. Thing learned simple sensorimotor policies first and then uses them as abstract actions in a behavioral hierarchy. These policies are represented as nondeterministic Finite State Automata (FSA) where actions are confined to belong to well-defined subsets of the control basis that are provably consistent with policy specifications.

The first policy Thing learned was how to rotate in place(Huber and Grupen, 1997). By imposing a resource model on the actions available to our quadruped, the size of the functional state space was limited 2^{13} states with 1885 possible concurrent control actions from each of these states. A quasistatic stability constraint is implemented in the DEDS specification as a logical disjunction over four controllers that stabilize different tripod stances and certifies that at least one must be near equilibrium at all times. Together with another kinematic feasibility constraint, the number of relevant states was reduced to just 32 with an average of 157 available concurrent control options available from each state. These constraints are expressed as logical conditions on the pattern of equilibria in the active controllers. Any states (or actions that produce future states) that violate these constraints are disallowed. In addition to producing performance guarantees, constraint satisfaction has a dramatic influence on the complexity of learning.

Our walking platform acquired policies for rotation gaits in the structured search space described above in as little as 11 minutes, on-line, in a single trial using a standard Q-learning algorithm. The final policy appears in Figure 8.

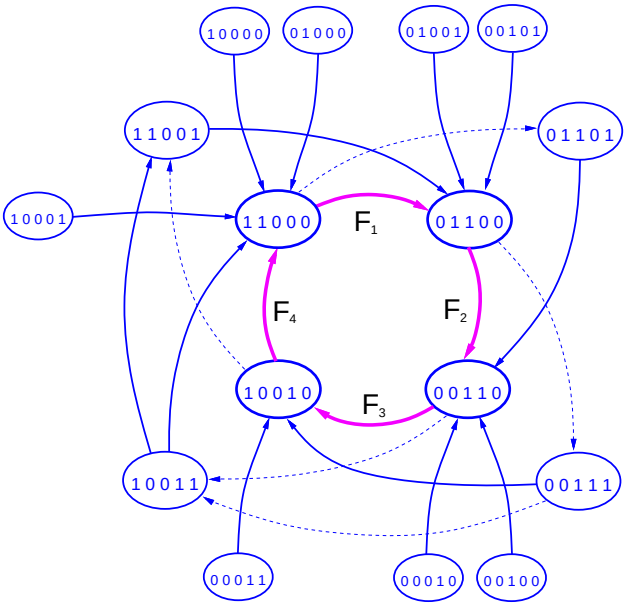

Figure 8: A Walking Gait - the policy responds to a variety of sensorimotor contexts. The central cycle has transition probabilities of greater that 95% in the training context.

Knowledge of the motor synergies involved in rotating under varying conditions significantly improved the acquisition of other behavior. For instance, Thing learned to translate in roughly half the time given the prior rotate policy than it did without it, and ultimately, the average amount of translation per action was roughly double as well (Huber, 2001). We believe that this phenomenon is consistent with the kind of staged, sequential development that human neonates exhibit as observed by developmental psychologists like Piaget.

Once schemata for rotating and translating are in place, navigating in a cluttered environment can be formulated as a policy for deciding when to translate and when to rotate in response to observed obstacles. We demonstrated that Thing can find a path from point A to point B with no prior knowledge of the intervening obstacles using a forward-looking IR proximity detector to observe obstacles enroute and map them into an evolving configuration space. The locomotion plan follows a streamline in a har-

monic function path controller by selecting one of two temporally extended actions (ROTATE-GAIT, or TRANSLATE-GAIT) in a 4 state finite state automaton. The state is derived from a 2 bit "interaction-based" state descriptor. One bit describes the convergence status of the ROTATE-GAIT action, and the other describes the convergence status of the TRANSLATE-GAIT action with respect to the current path plan. The control knowledge represented in these two actions and this FSA are adequate for finding paths in any cluttered world provided that a path exists at the resolution of our configuration space map.

Policies built using this framework for perceiving state and executing control are very robust to changing conditions. Essentially the same policy is used when our 4-fingered Utah/MIT hand rotates a cylinder and screws in a light bulbs. A recent extension (Huber and Grupen, 2002),

4.4 ACQUIRE *Schemata*

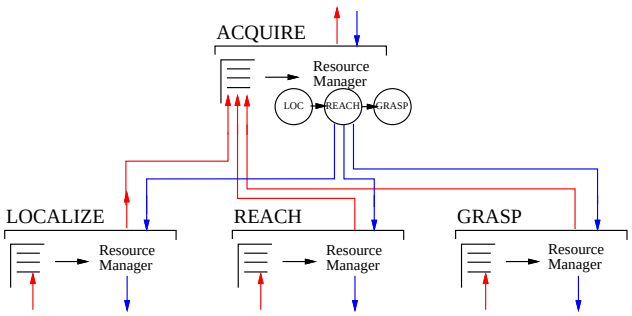

Figure 9: ACQUIRE - A Physical Schema for acquiring objects distributed about the environment.

By combining instances of the schemata already presented, more interesting behavior can be constructed. In this example, we use a special form of the LOCOMOTE schema called REACH - it can thought of as an appropriately parameterized LOCOMOTE that employs degrees of freedom in the arm to move fingers and tactile sensors to the object sought. Schemata for LOCALIZE, REACH, and GRASP are much more appealing constructs for programming ACQUIRE because they each encode a great deal of control knowledge and each will take actions based on the dynamics of the run-time environment to achieve their respective goals. The ACQUIRE schema may exploit state and context returned in each of the components to manage resources in the composite controller more effectively.

5. Conclusions

5.1 Software Infrastructure

There appears to be a great deal of similar structure common to the schemata we have presented. For instance, each of these examples uses local memory to estimate the control context from a time series of observations and a learning component to associate the categories observed in each domain into control decisions regarding actions and resources. The implementation of a truly autonomous developing robot will require that much attention is paid to a recursive and self organizing software structure. While processes and structure in these hypothetical software structures are shared, the control knowledge and policies for resource management will be specific to the domain of interaction represented by the schema.

5.2 Relativized Actions

The problem is that policies are acquired over time by addressing tasks in *particular* control contexts. Given enough experience, policies can approach optimal in the training set, but we would like to achieve a degree of abstraction that extends this policy to an entire class of behavior without having to train the agent uniformly over the entire domain. To scale well, lifelong learning tasks must use related experience to inform exploration during future learning tasks.

The control basis generates closed-loop controllers by sampling combinations of $\{ \Phi | \phi_i \in (F \times 2^{\Omega_s} \times 2^{\Omega_e}) \}$, where F is the set of value functions both native and derived, Ω_s is the stream of observable events ("sensor outputs") generated during training, and Ω_e is the "effector" designation. When a policy is acquired, it is expressed in the form of a sequence of control configurations drawn from this set.

Physical schemata focus specifically on the mixture and sequence of objectives F that collectively solve a particular class of tasks. From this view, each instance of a controller is a schema that has been decorated with sensor and effector resources to solve a particular element of the class. Physical Schemata form a declarative basis for this behavioral class that can be learned from every training episode drawn from this class.

5.3 Homomorphism and Reflection

Our mechanism for generalization generates opportunities to apply a policy developed in one context to another (related) context. We take our lead here, from techniques originally developed for finding minimal Markov models (Dean and Givan, 1997, Ravindran and Barto, 2001). These graph re-

duction techniques involve identifying topological structure in an MDP that can exploit *symmetry* (Popplestone and Grupen, 2000, Ravindran and Barto, 2001) to reduce the complexity of the model. We observe that the generalization of an existing policy can be defined by groups of homomorphisms. These transformations are used to hypothesize symmetries in the MDP and create reflections of existing policies in other (related) domains. For example, homomorphic transformations can involve permuting the resources employed in an otherwise constant control policy. Such a transformation preserves the sequence and combination of objectives underlying the task but executes the schema using a different suite of sensor and motor resources.

5.4 Human-Robot Interaction (HRI)

Parents have a great deal of input into the developmental growth of their children. Interfaces will be required to accomplish this for robot systems as well. For example, we can try to *explain* teleoperator inputs by searching for elements of the control basis that produce similar actions in this context. Further exploration can be focused exclusively on such plausible controllers to generate an internal representation of the teleoperated task expressed in terms of value functions native to the device. Perhaps constraints expressed in the DEDS-based task description and/or the resource model can be used to "teach" the system how to explain new concepts incrementally as in classical approaches to *shaping* and *maturation*.

Additional research in this area is likely to yield techniques for developing physical schema derived from teleoperator input that express the policy using native value functions. A generalization to such an approach can be used to negotiate a shared meaning between robots, humans, and the adaptive interface, each of which employ customized internal models tuned to their particular task that ultimately describe the same physical phenomena.

Acknowledgements

The work described herein was supported in part by the NSF (CDA 9703217), DARPA-MARS (DABT63-99-1-0004), NASA (NAG9-1445#1), and the University of Massachusetts. Any opinions, findings, conclusions, or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

Aronson, A. (1981). *Clinical Examinations in Neurology*. W.B. Saunders Co., Philadelphia, PA.

Barto, A., Bradtke, S., and Singh, S. (1993). Learning to act using real-time dynamic programming. Technical Report CMPSCI Technical Report 93-02, Computer Science Department, University of Massachusetts, Amherst, MA.

Bernstein, N. (1967). *The Coordination and Regulation of Movements*. Pergamon Press, Oxford, England.

Berthier, N., Clifton, R., McCall, D., and Robin, D. (1999). Proximodistal structure of early reaching in human infants. *Experimental Brain Research*, 127:259–269.

Bril, B. and Brenière, Y. (1992). Postural requirements and progression velocity in young walkers. *Journal of Motor Behavior*, 24(1):105–116.

Coelho, J. (2001). *Multifingered Grasping: Haptic Reflexes and Control Context*. PhD thesis, University of Massachusetts, Amherst, MA.

Coelho, J. and Grupen, R. (1997). A control basis for learning multifingered grasps. *Journal of Robotic Systems*, 14(7):545–557.

Dean, T. and Givan, R. (1997). Model minimization in markov decision processes. In *Proceedings of the AAAI*. AAAI.

Fiorentino, M. R. (1981). *A Basis for Sensorimotor Development - Normal and Abnormal*. Charles C. Thomas, Springfield, IL.

Gibbs, G. J. and Colston, H. L. (1995). The cognitive psychological reality of image schemas and their transforms. *Cognitive Linguistics*, 6(4):347–378.

Grupen, R., Coelho, J., and Souccar, K. (1992). Online grasp estimator: A partitioned state space approach. Technical Report CS Technical Report 92-75, CS Department, University of Massachusetts.

Huber, M. (2001). *Huber's Dissertation Title*. PhD thesis, University of Massachusetts, Amherst, MA.

Huber, M. and Grupen, R. (1997). A feedback control structure for on-line learning tasks. *Journal of Robots and Autonomous Systems*, 22(3-4):303–315.

Huber, M. and Grupen, R. (2002). Robust finger gaits from closed-loop controllers. In *IEEE Conference on Robotics and Automation*, Laussane, Switzerland. IEEE.

Huber, M., MacDonald, W., and Grupen, R. (1996). A control basis for multilegged walking. In *Proceedings of the Conference on Robotics and Automation*, Minneapolis, MN. IEEE.

Johnson, M. (1987). *The Body in the Mind*. University of Chicago Press.

Kuypers, H. (1981). *Anatomy of the Descending Pathways*. American Physiology Society, Bethesda, MD.

Lakoff, G. (1984). *Women, Fire, and Dangerous Things*. University of Chicago Press.

Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press.

Mandler, J. M. (1988). How to build a baby: On the development of an accessible representational system. *Cognitive Development*, 3:113–136.

Mandler, J. M. (1992). How to build a baby: Ii. conceptual primitives. *Psychological Review*, 99(4):587–604.

McGeer, T. (1992). Principles of walking and running. *Advances in Comparitive and Environmental Physiology*, 11.

McGeer, T. (1993). Dynamics and control of bipedal locomotion. *Journal of Theoretical Biology*, 16:277–314.

Mussa-Ivaldi, F., Bizzi, E., and Giszter, S. (1991). Transforming plans into action by tuning passive behavior: A field approximation approach. In *Proceedings of the 1991 International Symposium on Intelligent Control*, pages 101–109, Arlington, VA. IEEE.

Özveren, C. and Willsky, A. (1990). Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, 35(7):797–806.

Piaget, J. (1952). *The Origins of Intelligence in Childhood*. International Universities Press.

Piaget, J. (1954). *The Construction of Reality in the Child*. Basic Books.

Popplestone, R. and Grupen, R. (2000). Symmetries in world geometry and adaptive system behaviour. In *Proceedings of the 2nd International Workshop on Algebraic Frames for the Perception-Action Cycle (AFPAC 2000)*, Kiel, Germany.

Ramadge, P. J. and Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–97.

Ravindran, B. and Barto, A. (2001). Symmetries and model minimization in markov decision processes. Technical Report CMPSCI Technical Report 01-43, Computer Science Department, University of Massachusetts, Amherst, MA.

Rimon, E. and Koditschek, D. (1989). The construction of analytic diffeomorphisms for exact robot navigation on star worlds. In *Proceedings of the 1989 Conference on Robotics and Automation*, volume 1, pages 21–26, Scottsdale, AZ. IEEE.

Salisbury, J. (1982). *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Stanford University.

Schaal, S. and Sternad, D. (1998). Programmable pattern generators. In *Proceedings of the International Conference on Computational Intelligence in Neuroscience*, pages 48–51.

Sobh, M., Owen, J., Valvanis, K., and Gracani, D. (1994). A subject-indexed bibliography of discrete event dynamic systems. *IEEE Robotics & Automation Magazine*, 1(2):14–20.

Thelen, E. and Smith, L. (1994). *A Dynamic Systems Approach to the Development of Cognition and Action*. MIT Press, Cambridge, MA.

Williamson, M. (1999). Neural control of rhythmic arm movements. *Neural Networks*, 11(7-8):1379–1394.

Wolff, P. (1991). Endogenous motor rythms in young infants. In Fagard, J. and Wolff, P., (Eds.), *The Development of Timing Control and Temporal Organization in Coordinated Action*. Elsevier Science.