

Evolvable Mechanics: Hardware tools for evolutionary robotics

Bill Bigge. Inman. R. Harvey

Abstract—Embodying robot morphologies evolved in simulation can present serious problems for an engineer when translating simplified simulated mechanisms into working devices, often drawing on pre-existing component parts to provide specific functions, to reduce construction times and to cut costs. Actuators for generating motion in articulated joints are key components of any robot and a vast range of compact, low cost devices are available, but their behavior is based on generating positions and velocities by ironing out perturbations.

In simulation it is easy to create variable compliance actuation schemes where damping and spring forces used to guide motion can be subject to evolutionary search and aid the generation of efficient adaptive behavior. We present an experimental design for a cost effective actuator module with software controlled compliance that can approximate arbitrary mechanical spring damping systems, with parameters that are software configurable at run time. The benefits of this software controlled compliance and the versatility of our embedded control system are demonstrated in a minimal walking robot. The potential value and inherent limitations of this device as a tool for evolutionary robotics, where software controlled compliant properties can form part of an evolutionary search, and in the wider field of autonomous robotics, are discussed.

I. INTRODUCTION

A common feature of any robot with articulated joints are actuators that allow for controlled movement. The dominant control methodology for these types of actuator is that of position and velocity control where the actuator is required to maintain user specified angles and transition between angles as specified velocities. In order to achieve this behaviour the actuator's control system must try and minimise the effects of external perturbations on its motion to create a device that is as rigid as possible with respect to external influences.

This methodology comes out of the need for industrial robots to produce accurate and repeatable movements within a constrained, known environment but the demands on autonomous robots are different and research over the last few decades has demonstrated that sensing and reacting to external perturbations can aid the generation of efficient adaptive motion, for example in passive dynamic walking or adaption to uneven terrain, [1-4] as well as reducing the potential for harm caused during encounters with objects in the environment and during human machine interactions[5-7]. Conventional rigid actuation schemes do not allow these physical interactions to occur and instead impose prescribed

trajectories on a mechanism. Actuators that accept external influences allow trajectories to emerge that are a product of both the actions of the embodied mechanism and the situation they are in. In this context actuators are not only output devices but form a bi-directional interface to the environment. Creating actuators that are both inputs and outputs - sensors of actuation - is an important step towards realising more complete and fully embodied autonomous agents and helps fulfil many of the design principles for 'Fungus Eaters' outlined by Rolf Pfeifer [8].

In evolutionary robotics a genetic algorithm is most commonly employed to evolve the artificial brain and nervous system of the robot, which then controls a morphology that is largely fixed. This is in contrast to biological evolution where brain and body both co-evolve. Attempts to evolve robot morphology have typically been carried out entirely in simulation [9, 10] with some exceptions, for example in evolving sensor morphology on real robots [11] or in using rapid prototyping tools or modular construction to translate simulated morphology into real robots [12, 13]. To date there has been little attempt to evolve the mechanical characteristics of passive and active articulated joints in simulation or in embodied systems, but doing so can radically extend the possibilities for an evolutionary search [9].

A number of different actuator systems have been developed that are designed to exhibit various forms of controlled compliance, and have potential to allow joint compliance to form part of an evolutionary search. Two dominant approaches are mechanically controlled stiffness and active force control. The former of these approaches relies on actuating a joint with a pair of non-linear springs whose tension can be varied to create a controlled equilibrium point between variable stiffness springs [14-17]. The latter approach relies on sensing torque differences between the load and the drive, and employing a control loop to maintain a user-specified torque level, resulting in control over the force that the actuator will generate. Two distinct active force control methods are joint torque control [18] where the torque sensing element is rigid, and the Series Elastic Actuator [19] that relies on an instrumented spring between the drive and the load.

Whilst mechanically controlled stiffness can provide the most energy efficient method of controlling joint stiffness it is limited to providing a movable equilibrium point with variable stiffness, whereas active methods offer more versatile control at the expense of energy efficiency. Although some of these devices have been developed into commercial products for robotics research they remain

costly and are usually aimed at high specification, large scale robots. For the researcher wanting to explore behaviours that exploit dynamic interactions between morphology and environment in small scale, low cost robots there are few if any viable off-the-shelf actuators capable of controlled compliance, and none, before the present work, that have been appropriate for exploiting an evolutionary robotics design methodology.

A key element in the design of a modular compliant actuator is that of utility. Conventional rigid actuators deployed in low cost research robots provide integrated control so the researcher can send position and velocity commands and receive back angle sensor data. The researcher does not need to invest time designing and programming the motion control hardware and as a consequence they do not risk introducing bugs into the low level control software. The hardware wraps up all the functions needed to produce the range of required behaviours.

For conventional actuators these behavioural requirements are well defined, one wants to specify angles and velocities, but in a system with controlled compliance the requirements are more diverse but can be loosely described in terms of some transfer function between the measured angle and velocity, and the force output. Encapsulating this range of behaviours in some formalised control architecture can provide some of the benefits listed above, namely that the researcher can concentrate on specifying behaviour parameters rather than debugging embedded software.

This paper presents an attempt to create an actuator and control system that can provide some of this functionality by allowing behaviour to be defined in terms of emulated spring and damping forces. We will start by describing the basic mechanical design of our actuator, followed by a description of the high level control system. The utility of this control system and some associated features of the hardware will be demonstrated with two examples, a mechanical oscillator based on variable spring and damping forces with hysteresis, and a minimal walking robot that uses perturbation to trigger a stepping reflex. We will end with a discussion on the limitations of our system and the ways in which our approach can be applied to the field of evolutionary robotics.

II. THE PROGRAMMABLE SPRING ACTUATOR

Our system is based on an existing design for a force controlled actuator, the Series Elastic Actuator [19]. We have combined it with extra control and sensing elements to produce a compact integrated modular device that can be programmed to approximate a wide variety of compliant mechanical joints, all based on variable spring and damping forces. By developing a modular device with known and controllable compliance properties, which can be produced at relatively low cost, we hope to make certain aspects of evolutionary robotics easier. Joint properties can be evolved in simulation within the parameters specified by the actuator hardware and immediately implemented in a real robot, or

evolution can be applied directly to an embodied robot.

Fig. 1 shows a schematic of our prototype, the Programmable Spring Actuator [20, 21], and consists of a motor and gearbox whose output is coupled to a shaft via a torsion spring, instrumented with a potentiometer to measure the angular deflection of the spring. Torque differences between the drive and the output produce a linear and proportional deflection of the spring allowing the torque to be directly measured by a microcontroller. A simple control loop then drives the motor to maintain a specified spring deflection, reproducing the function of a Series Elastic Actuator. The mechanical output is rotary and terminated in a flat disk at each end to facilitate easy construction of articulated joints.

A second angle sensor and control circuit allow the angle of the output shaft to be measured, and used to generate force commands for the force control module. This control unit also provides external electronic interfaces designed to facilitate control over the actuator, to return sensor data and to allow signalling between groups of actuators.

A key component of this design is the control system that translates angle and angle velocity data into force commands, and handles communication with the outside world. It is structured in an object-orientated fashion where an actuator can contain up to eight objects that encapsulate parameters for controlling spring and damping forces, and a number of other control elements relating to external communications.

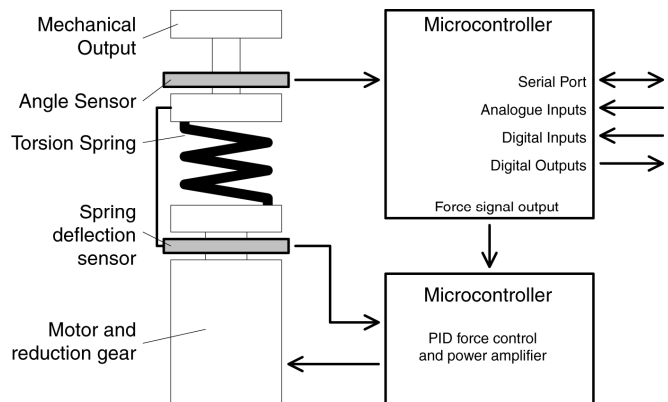


Fig. 1. Programmable Spring Actuator schematic.

Each object, called a profile group, can be placed in control of the actuator and contains methods for defining when it passes control to a different object. In essence the user can employ a single profile group to specify an arbitrary continuous spring damping system that the actuator will try and approximate, or employ multiple profile groups to create more complex systems that generate oscillations, hysteresis and antagonistic action.

The core elements of a profile group are as follows:

Force profile – A look-up table of 64 values describing spring forces across the angular range. A linear interpolation algorithm is used to translate these to the 1024 angles that can be measured by the hardware.

Damping profiles – A pair of 64 element look-up tables that describe damping forces across the angular range in the same fashion as the force profile. Each damping profile applies to one direction of motion only.

Modulating variables – A set of methods for adjusting the scale and position of the force and damping profiles with respect to the angle, and which allow the mechanical equilibrium points and the spring and damping constants defined by the profiles to be adjusted, generating motion and controlling stiffness.

Threshold checking – A set of user defined thresholds that can be applied in pairs to any actuator variable. A pair of positive and negative thresholds can evaluate a variable, for example the angle or velocity, and indicate if either threshold has been passed. Pairs of thresholds can also be switched on or off by other conditional events.

Profile switching – A module that evaluates conditional events like thresholds that, if active, will pass control to an alternate profile group.

Profile summing – A method of summing the force output of a pair of profile groups, allowing two sets of independently adjustable spring dampers to operate in an antagonistic fashion.

External communications – A set of modules that control the hardware communication interfaces. Packets of user defined data can be transmitted serially, and incoming variables mapped to modulating variables. Digital inputs can be mapped to conditional triggers, for example to inhibit thresholds or switch profiles, and digital outputs can be mapped from conditional events, for example to indicate the status of a threshold to an external device.

All of these parameters can be set with a graphical user interface running on a host computer. Spring and damping profiles can be defined by literally drawing them with the mouse or by generating them algorithmically. Other elements are configured by calling up menus and selecting parameters from lists. When complete a configuration consisting of multiple profile groups can be downloaded to the actuator.

A primary goal in the design of this control system was to create a device that could be programmed, or specified genetically using an evolutionary algorithm, to approximate the behaviour of various compliant mechanical systems and which offered a degree of versatility in the way groups of devices could be used when constructing robotic systems. The various external communication channels provided by each actuator combined with the profile group control system can be used to embed complex reactive behaviours within groups of actuators without the need for an additional central controller, whilst preserving the ability to exert direct control when required. Although the most versatile approach is simply to provide the hardware, and allow the user to code the individual actuator’s software, this can be a time consuming process. By providing a structured control system we hope to facilitate the rapid development of experimental compliant robot morphologies by removing the

need to programme and debug individual hardware elements. With our system the user can ‘sketch’ a mechanical system and with a suitable fitness function an evolutionary algorithm can then optimise the design.

III. CONFIGURING AN OSCILLATING SYSTEM

To demonstrate how an actuator can be configured and how this translates into behaviour we produced a simple oscillating system that included some conditional damping across select angles.

To produce the oscillator we used a pair of profile groups, illustrated in Fig. 2. The first group defines a series of spring forces across most of the angular range with an angular threshold set towards one end. This spring force will tend to drive the output towards the threshold. An irregular series of damping forces are also defined which will modulate the speed of the actuator as it travels towards the threshold. These damping forces apply only when the output is travelling in this direction. The profile switching unit is also configured to evaluate the status of the threshold and if passed it will transfer control to the second profile group.

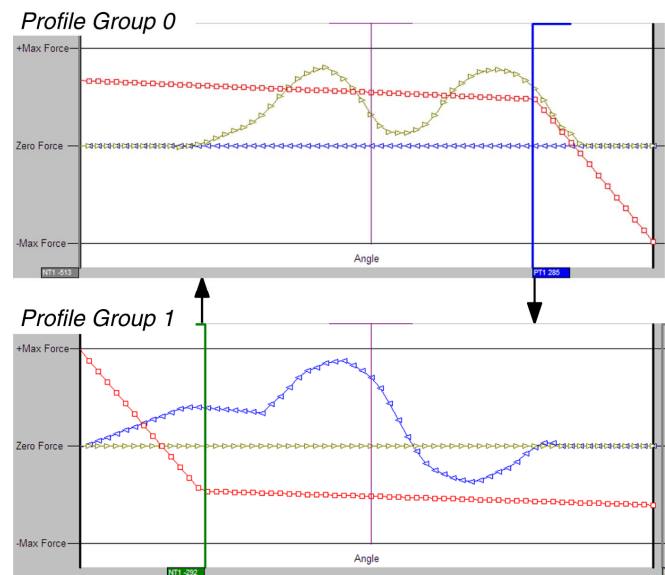


Fig. 2. Configuring an oscillator. The force profile is indicated by a red line with square symbols at each data point (each location in the look-up table) with the two damping profiles represented by blue and green lines, each with arrows that denote the direction of motion to which the damping applies. For forces, the vertical axis represents the force value but for damping the vertical axis represents the magnitude of damping. Negative values for damping are allowed and will amplify rather than inhibit motion.

The second profile group is configured in a similar way but with a spring profile that pushes the output in the opposite direction towards another threshold. Damping for this direction of motion is also defined in an arbitrary fashion and the profile switching unit will pass control back to the original group when the output goes past this second threshold.

IV. VISUALIZING FORCE AND DAMPING

The utility we created for configuring the actuators contains a method of visualising how the force and damping for each group will translate into forces. We compute the force value that the Series Elastic Actuator will generate in advance and display this as a two-dimensional coloured grid of forces for every angle and velocity that the actuator can achieve. Red and blue colours are used to distinguish the sign of the force with force magnitude denoted by brightness.

Force and velocity data from the actuator can be sampled and transmitted using the inbuilt communications system. This data can be interpreted by the configuration utility and displayed as a point or a trajectory on the force surface. With our oscillator configuration complete it is downloaded to the actuator which was mounted on a bench and had a short lever attached to the output. The lever had a small mass attached and the whole system was mounted so the lever would move about the vertical axis and as such was not influenced by gravity. A start signal sent to the actuator and the resulting trajectory is shown in Fig. 3. Because the actuator is using a pair of profile groups to generate the behaviour there are actually two different force surfaces that apply to the system. In Fig. 3 we have shown the trajectory of the actuator across a composite force surface. This is constructed from the two profile groups and shows the areas of the force surface from each group that are active for any angle and velocity.

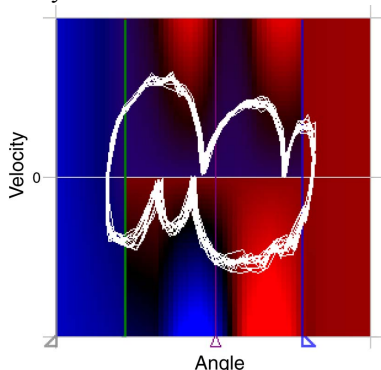


Fig. 3. Oscillator trajectories across a force surface. Angle and velocity data captured as the oscillator runs is displayed as a trace across the surface. The force surface is a composite of the relevant parts of two profile groups and represents the forces that would be generated for every angle and velocity. The sign and magnitude of the force is represented by the colour and brightness with blue denoting forces that would push the angle in a positive direction (towards the right).

Unlike a conventional rigid actuator the trajectory produced here is the product of a complex spring damping system, not an imposed rule. With a more conventional system we could instruct the actuator to generate this trajectory but any perturbations introduced to the mechanical output would be ironed out to maintain the prescribed motion. With this compliance-based actuator perturbations can have a significant influence on motion so the resulting trajectory emerges from the interaction between forces generated by, and encountered by, the actuator. Optimising this relationship between the actuator configurations and

resulting trajectories for a specific task can be challenging for a human designer, particularly when using several actuators, but refining these complex relationships is an ideal task for an evolutionary approach.

V. A ROBOT WITH AN EMBEDDED WALKING REFLEX

Creating robotic walking platforms that can react in useful ways to external influences can be a taxing design task. The ways in which a platform can sense and react to the physical environment are complex, particularly when working outside the paradigm of rigid actuator behaviour.

In order to demonstrate how our hardware can be used to create novel behaviours we produced a minimal walking robot that could be programmed, by adjusting the compliant properties of the leg joints, to walk when pushed. The resulting behaviour is only possible because the leg joints and their controllers allow external forces to influence their behaviour, which can then feed back into forces that generate motion. The various profile groups that define the overall behaviour were carefully evolved by hand but they could also be put under the control of a genetic algorithm.

The example robot has a morphology that is fixed in terms of the physical arrangement of actuators, but the compliant properties of the actuators are under software control. With an evolutionary approach we could extend the range of parameters that the genetic algorithm was able to define to include the lengths of each leg joint and the relative orientation of those joints to each other. In the following section we will discuss how this class of actuator would fit into an evolutionary methodology.

We began by constructing a single leg consisting of a pair of actuators. One operated the shoulder and could swing the leg forwards and backwards whilst the second operated the knee to lift the foot off the ground. Three profile groups were defined for the shoulder actuator with the first default group defining an equilibrium point between a pair of soft springs that would keep the shoulder in a central position whilst offering a degree of compliance. This group also included a pair of angular thresholds situated either side of the equilibrium point, control would pass to one of the other two profile groups when either of these thresholds were passed.

These alternate groups each defined a second equilibrium point bounded by stiffer springs, with a degree of damping in one direction of motion and another threshold that will return the actuator to the default group when passed. When the shoulder is active and perturbed beyond one of the thresholds the alternate group will exert a force to pull the shoulder in the opposite direction before returning to the default equilibrium point. An additional set of parameters for the two alternate groups would trigger a state change on a digital output for as long as either group was active.

The knee actuator used a pair of profile groups with the default consisting of an equilibrium point bounded by an asymmetric spring damper designed to lift the robot's body off the floor by pushing the leg down. A second group

defined a new equilibrium point between a pair of springs designed to lift the leg off the floor. A digital input was then used to detect the state of the shoulder actuator and lift the leg when the shoulder was using either of its alternate profile groups. A diagram illustrating the force and damping profiles used for the knee and shoulder is shown in Fig. 4.

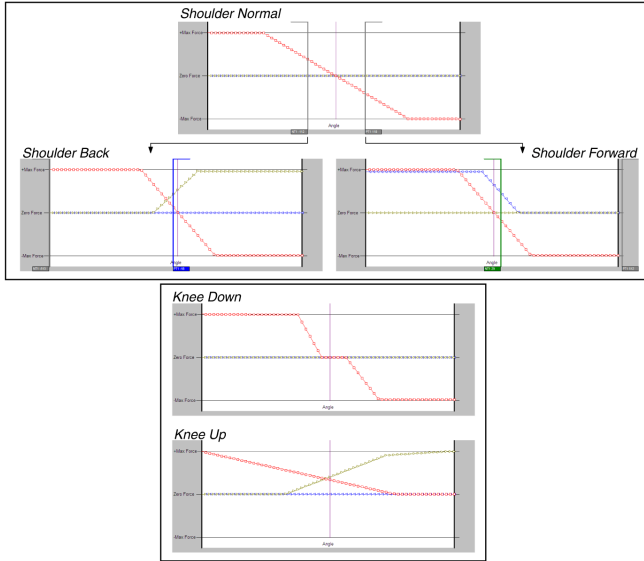


Fig. 4. A reflex leg configuration. Three profile groups for the shoulder are shown at the top and the two for the knee are on the bottom. The ‘Knee Down’ profile is always active when the ‘Shoulder Normal’ profile is active.

The result of this system was a leg which would lift its foot from the ground and take a step when perturbed beyond the angular thresholds. To test it we attached the leg to a wheeled platform and pushed it along the ground to trigger the stepping reflex. The changing knee and shoulder angles were captured and are shown in Fig. 5.

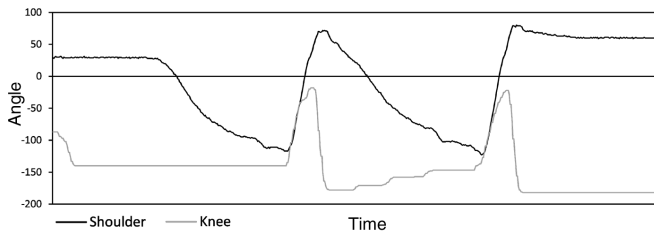


Fig. 5. Shoulder and knee motion as the robot walks. The angles of the two actuators plotted as the leg stepping reflex is triggered.

Next we constructed a second identical leg and attached it to a platform along with the first leg. We added a pair of stabilising wheels to the platform to remove the need for more than two legs and added wheels to the end of each leg to reduce sliding friction when the leg pushed down onto the floor. Our prototype actuators were primarily designed to develop and demonstrate the control system and unfortunately lacked sufficient torque to properly lift the weight of the robot. We added a counterbalance to the rear of the robot which, in conjunction with the wheeled legs, removed the need for us to construct new actuators with

more powerful motors.

In order to prevent both legs taking a step at the same time we updated the configuration of the shoulder actuators so that the two thresholds that triggered the reflex would be inhibited if a signal was detected on a digital input, and another digital output would become active when the reflex was triggered. By cross-connecting wires between the two shoulder actuators we could ensure that the first leg to take a step would inhibit stepping in the second leg until it had completed its step. The finished robot is shown in Fig. 6 and when pulled along with a leash it would obediently follow with a co-ordinated, if reluctant, walking gait. This basic reflex architecture could be further expanded to incorporate more legs with an additional degree of freedom to produce a unidirectional walking platform with no supporting wheels.

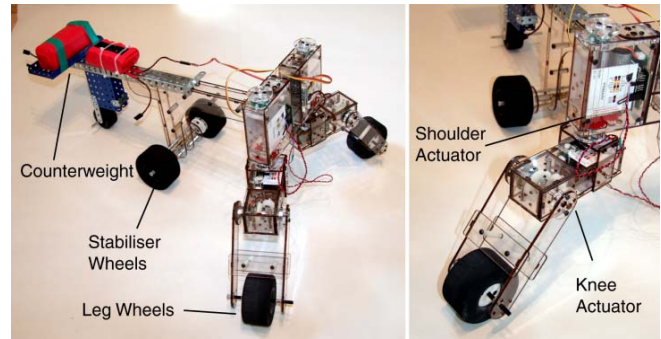


Fig. 6. A minimal walking robot.

A series of videos demonstrating the robot along with demonstrations of other actuator behaviours can be found at the following URL:

www.informatics.sussex.ac.uk/users/wb23/PGS/Videos.html

VI. DISCUSSION

The robot walker serves to demonstrate a number of things; firstly that novel reactive behaviours can be produced if actuation hardware is available that can generate appropriate compliant behaviour. It also demonstrates how our approach of developing a low-cost compliant actuator with a sophisticated, user-orientated control system can be used to rapidly construct robotic platforms capable of producing dynamic reactive behaviours that are only possible when control of joint compliance is available. In our example the behaviours were carefully evolved manually over the course of a few hours, for the evolutionary robotics community we want to stress how this opens up new possibilities; many of the parameters that define the compliance of the joints could equally be put under the control of a genetic algorithm to be evolved either in simulation or on an embodied platform.

Our control system presents some limitations and one relevant to the demonstration robot is that there is no effective way of modulating the gait to compensate for different rates of perturbation. In other words the stepping reflex remains the same regardless of how fast you pull the robot. Refining the profile group system to try and accommodate all possibilities for constructing useful robotic

behaviours could become a pointless task so we have concluded that the most appropriate system is one that allows the actuator to approximate a diversity of mechanical joints, including models of joints actuated antagonistically with biological muscles, alongside a robust and structured set of communication interfaces to allow easy integration into larger systems. Effectively the actuator would contain a minimal physical simulator whose output would determine its mechanical forces and velocities.

The current control system includes a mode of operation that allows a pair of profile groups to operate simultaneously by summing their respective force and damping profile outputs. This enables the actuator to emulate the behaviour of antagonistic actuation. There may be value in enhancing this feature of the system to provide emulation of some specific models of biological muscles [22]. The damping profiles can also be employed as a method of reproducing the effects of variable friction in a joint, allowing the actuator to emulate the effects of mechanical wear, damage and contamination.

The microcontroller that we used for our prototype is underpowered for the task but proved adequate for our purposes. More modern devices offer an order of magnitude more performance and, if used in a more refined design, would provide some spare compute capacity. Spare capacity in a modular design like ours could be used to distribute certain higher-level control tasks across groups of actuators. Options we have considered that make use of this capacity to usefully augment the control system are the inclusion of a minimal neural network which could serve as a pattern generator for controlling motion. A more sophisticated approach would be to allow the researcher to develop their own code which can be downloaded as part of the actuator's configuration. This would help preserve the correct, bug-free functionality of the mechanical emulation whilst offering more flexibility.

The potential of using the Series Elastic Actuator coupled with our control system to approximate or emulate mechanical joints is appealing but there are some limitations imposed by the hardware that restrict the range of behaviours that can be generated. Control loop latency is a problem common to the stability of any actuator, but can be mitigated to a degree by using the angular velocity to predict what the actual angle will be when a new force value has been computed. Forces produced by the hardware are also not instantaneous so the actual force bandwidth will influence behaviour. With Series Elastic Actuators the relatively large degree of mechanical compliance in the spring results in a force bandwidth that reduces as the force magnitude increases. The electric motors that drive the actuator also have limitations that influence performance; in particular the torque they generate tends to have an inverse relationship to their velocity. When compared to a real spring under compression the actuator might be capable of generating the same force when static but be incapable of matching the velocity that the spring will generate as it

relaxes when released. In this instance the actuator would only be able to reproduce the behaviour of a damped spring.

Active compliance generators also tend to be poor at generating rigid mechanical behaviour and our current system is limited to reproducing mechanical behaviours based on springs and dampers. Creating the equivalent of a hard mechanical stop is not possible with the system we have designed but may be desirable in some circumstances, for example to create locking knee joints in bipeds. It is possible to design additional mechanical elements to augment the existing device which would provide software movable mechanical stops, but this increases the complexity and weight of a device.

Active compliance devices can emulate low impedance passive joints but they do so by consuming power. In walking robots, for example those that rely on passive movement as part of the gait, the active emulation of passive compliance can reduce efficiency. For applications like these there may be good reasons to consider augmenting the actuator design to include some form of actuated clutch. This would physically decouple the actuator's output from the drive to produce a truly passive joint.

Although some of these performance issues can be addressed, or at least reduced by careful decisions during design, some of them represent inherent limitations of actuation technology and are better dealt with by making these limitations clear to the researcher when actuators are being configured and used. The force surface visualisation we developed is a useful way of visualising the target behaviour of an actuator but it does not currently take into account these hardware limitations. It would be possible to enhance this visualisation by computing force values that include compensation for the control loop latency as well as accounting for the torque reduction that occurs as the motor velocity saturates.

The primary focus in the development of our actuator and control system has been to allow mechanical behaviours to be quickly and intuitively designed but the resulting configurations contain a large amount of data. Using evolutionary algorithms directly to create configurations, particularly in robots with many actuators, would be challenging as the resulting search space may consist of many thousands of dimensions. Understanding how evolutionary algorithms could make the best use of the potential behaviours available with these devices will be vital in the development of more refined control systems. An example would be if a genetic algorithm were applied to the look-up tables that define force and damping. For a single actuator there are 192 variables to define so for a robot with just eight actuators the genetic algorithm would be searching a space with over 1500 dimensions. Genetic algorithms provide an excellent way of searching large dimensional spaces like this but searches can also be hampered if a large search space contains few viable parameters. It is not immediately obvious what range of configurations that can be produced for our actuator's are of

benefit, or an impediment, to an evolutionary approach. Generating these tables algorithmically, with the algorithm's parameters subject to evolution, would present a drastically smaller search space and may be a fruitful approach, but it would also constrain the types of mechanical behaviour that could evolve. The question this poses, and which remains unanswered, is what limits might be placed on the force and damping profiles that usefully constrain the available parameters?

VII. CONCLUSION

As a feature in the function of robotic actuators, controlled compliance can be of great value when trying to produce robust and adaptive behaviour in embodied robots. As a tool for evolutionary robotics, devices like our Programmable Spring Actuator offer the potential of putting certain morphological aspects of a real robot under the control of a genetic algorithm without requiring the manual construction or re-engineering of complex mechanisms. Joint properties, constrained to remain within parameters achievable by the actuators, can be evolved in simulation and transferred with confidence to real hardware, or evolutionary algorithms can be applied directly to an embodied platform.

Force-controlled actuators, a core component of our system, have performance limitations that need to be understood and explicitly quantified if devices like ours are to act as a useful tool in evolutionary robotics. Our existing prototype hardware and control system served as a test bed for developing a general purpose control system but the design still requires refinement in terms of performance and fidelity. What it represents is a first step towards the development of reconfigurable and evolvable mechanical modules. Much work still remains to refine the general design into something that will be of genuine value to the field of evolutionary robotics, but we have demonstrated how functional modular actuators with software controlled compliance, supported by intuitive user orientated tools, can be used to rapidly construct novel robotic morphologies and behaviours where the adjustable mechanical properties of joints form an essential component of the behaviour.

REFERENCES

- [1] T. McGeer, "Passive dynamic walking," *Int. J. of Robotics Research*, vol. 9, pp. 62-82, 1990.
- [2] E. D. Vaughan, E. A. Di Paolo, and I. R. Harvey, "The evolution of control and adaptation in a 3D powered passive dynamic walker," in *Proc. of 9th Int. Conf. on the Simulation and Synthesis of Living Systems*, Tremont Boston Hotel, 2004, pp. 139-145.
- [3] H. Kimura, Y. Fukuoka, and A. H. Cohen, "Adaptive Dynamic Walking of a Quadruped Robot on Natural Ground Based on Biological Concepts," *Int. J. of Robotics Research*, vol. 26, pp. 475-490, 2007.
- [4] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient Bipedal Robots Based on Passive Dynamic Walkers," in *Science Magazine*, vol. 307, 2005, pp. 1082-1085.
- [5] R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. Williamson, "The Cog Project: Building a Humanoid Robot," in *Computation for Metaphors, Analogy and Agents*, vol. 1562, C. Nehaniv, Ed.: Springer-Verlag, 1998, pp. 52-87.
- [6] M. M. Williamson, "Control of Rhythmic Arm Movements," *Neural Networks Special Issue on Neural Control of Movement*, vol. 11, pp. 1379-1394, 11 October 1998.
- [7] M. M. Williamson, "Rhythmic Robot Arm Control using Oscillators," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 1998*, Victoria, BC, Canada, 1998, pp. 77-83.
- [8] R. Pfeifer, "Building "Fungus Eaters": Design Principles of Autonomous Agents," in *Proc. of 4th Int. Conf. on Simulation of Adaptive Behavior*, Cape Cod, Massachusetts, 1996, pp. 3-12.
- [9] J. C. Bongard and C. Paul, "Making evolution an offer it can't refuse: Morphology and the extradimensional bypass," in *Proc. of 6th European Conf. on Artificial Life*, Prague, CZ, 2001, pp. 401-412.
- [10] J. C. Bongard and R. Pfeifer, "Evolving complete agents using artificial ontogeny," in *Morpho-functional Machines: The New Species: Designing Embodied Intelligence*, R. Hara and R. Pfeifer, Eds.: Springer-Verlag, 2003, pp. 237-258.
- [11] D. Cliff, P. Husbands, and I. Harvey, "Evolving visually guided robots," in *Proc. of 2nd Int. Conf. on the Simulation of Adaptive Behaviour*, Honolulu, Hawaii, 1993, pp. 374-383.
- [12] H. Lipson and J. B. Pollack, "Evolving Physical Creatures," in *Proc. of 7th Int. Conf. on Artificial Life (ALIFE7)*, Portland, Oregon, 2000, pp. 282-287.
- [13] I. Macinnes and E. A. Di Paolo, "Crawling out of the simulation: Evolving real robot morphologies using cheap, reusable modules," in *Proc. of 9th Int. Conf. on the Simulation and Synthesis of Living Systems*, Boston, 2004.
- [14] R. Van-Ham, "Compliant Actuation for Biologically Inspired Bipedal Walking Robots," PhD Thesis. *Department of Mechanical Engineering*. Brussels: Vrije Universiteit Brussel, 2006, p. 197.
- [15] B. Verrelst, F. Daerden, D. Lefeber, R. Van Ham, and T. Fabri, "Introducing Pleated Pneumatic Artificial Muscles for the actuation of legged robots: a one-dimensional set-up," in *Proc. of 3rd Int. Conf. on Climbing and Walking Robots*, Madrid, Spain, 2000, pp. 583-590.
- [16] J. W. Hurst, J. Chestnutt, and A. Rizzi, "An Actuator with Physically Variable Stiffness for Highly Dynamic Legged Locomotion," in *Proc. of ICRA '04 2004 IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 4662- 4667.
- [17] G. Tonietti, R. Schiavi, and A. Bicchi, "Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction," in *Proc. of 2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 526 - 531.
- [18] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer, "A new generation of torque controlled light-weight robots.," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 3356 - 3363.
- [19] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, 1995, pp. 399-406.
- [20] B. Bigge and I. R. Harvey, "Programmable Springs: Developing Compliant Actuators for Autonomous Robots," in *Proc. of Towards Autonomous Robotic Systems (TAROS) 2006*, University of Guildford, Surrey, U.K., 2006, pp. 13-18.
- [21] B. Bigge and I. R. Harvey, "Programmable springs: Developing actuators with programmable compliance for autonomous robots," *Robotics and Autonomous Systems*, vol. 55, pp. 728-734, September 2007.
- [22] A. V. Hill, "The heat of shortening and the dynamic constants of muscle," *Proc. of the Royal Society of London. Series B, Biological Sciences*, vol. 126, pp. 136-195, 1938.