

Carrying Heavy Objects by Multiple Manipulators with Self-Adapting Force Control

Jianwei Zhang, Markus Ferch and Alois Knoll

Faculty of Technology, University of Bielefeld, 33501 Bielefeld, Germany

Email: {zhang, imferch, knoll}@techfak.uni-bielefeld.de

Abstract

Hybrid force/position control of multiple robot arms to jointly carry an object in the whole overlapping working space is a non-linear problem. Firstly, the control parameters change with each robot's configuration, and secondly, the dynamic models of the other participating robots are not available if the robots are controlled distributedly. We adopt an approach of increasing learning through practising – the way a human would do it. Based on a B-spline network, each robot learns to find its optimal control parameters using on-line reinforcement signals. The robot controllers can exchange their learned parts of control surfaces in order to accelerate learning. We apply the approach to control three Puma-260 arms for jointly carrying relatively large and heavy objects. The successful experiments show that the automatic learning procedure converges in a short time and the resulting forces/torques can be reduced to the minimum.

1 Introduction

This work aims at automatic development of controllers which guides the cooperative motions of several robot arms based on their wrist-mounted force/torque sensors. The examination of this problem is motivated by applications such as transport of objects of large size or heavy weight and assembly of these objects. The motion of the robots must be controlled so that neither the objects nor the robots are damaged. The measured forces and torques on the manipulator wrists should be kept as small as possible.

A lot of research work on two-arm control employed the approach of modelling the manipulator dynamics and computing torques, e.g. [7]. If the dynamic model of a manipulator is *a priori* known, such an approach supplies an explicit physical interpretation of the motion-force process in form of differential equations. Therefore stability of such a control system can be analysed. Unfortunately for compliant motion control using industrial robots neither the parameters of the dynamic model of the robot are available nor is there a possibility to access the joint torques directly. Biological systems (several persons or ants transporting a heavy object) have neither an idea of their own nor of their partner's physical parameters.

Another important approach is the active multi-arm coordination based on forces and torques measured in the Cartesian tool space, e.g. [8, 1]. The compliant motion is realised by adjusting the stiffness of the controller using any type of PID control or a second-order low pass filter algorithm in frequency domain. Since there are numerous uncertainties in a real robot model like backlash, object internal tension forces, the imprecise modelling of real inertia parameters, etc., it is desirable that the stiffness of

such a compliant motion controller can be automatically adapted to different robots, objects and manipulation tasks.

Robot learning aims at generating robot software in an evolutionary way. Recently, some work using learning methods has been reported. Off-line supervised learning [4] utilised data from human demonstration, and a controller can be trained if the human instructor correctly demonstrated his skills. [5] discusses the training of a fuzzy-neural controller for position/force control through back-propagation and gives some simulation results. To train a controller for contour tracking based on force feedback, [6] used a neural network method. Reinforcement learning was applied in one-arm pendulum swing-up problem [3].

In this paper we present a practical approach for learning the relationship between the forces/torques and the compliant motion in the Cartesian space. For this purpose the B-spline model [10] is employed which can be classified into a neuro-fuzzy method. Its basic idea of partitioning input space with overlapping functions coincides with the CMAC [2] principle. Our early experiments with the B-spline model on numerous benchmark problems of modelling and also in mobile robot behaviour learning [11] and two-arm cooperation [9] have shown the good modelling capability of nonlinear relations, smooth output and rapid convergence of learning. In this work the principle of B-spline controllers is applied to the learning of compliant motion. Our experiment with a real robot system shows its suitability in a multi-arm cooperation task.

2 System Description

2.1 Set-up

We use three Puma-260 industrial manipulators hanging upside down (Fig. 1). Each manipulator is equipped with a force/torque sensor mounted between the last (sixth) joint and a pneumatic two-finger gripper. Each manipulator is controlled by an agent running on its own computer. The task which has to be performed is the translation of rigid, heavy objects with these three robots.

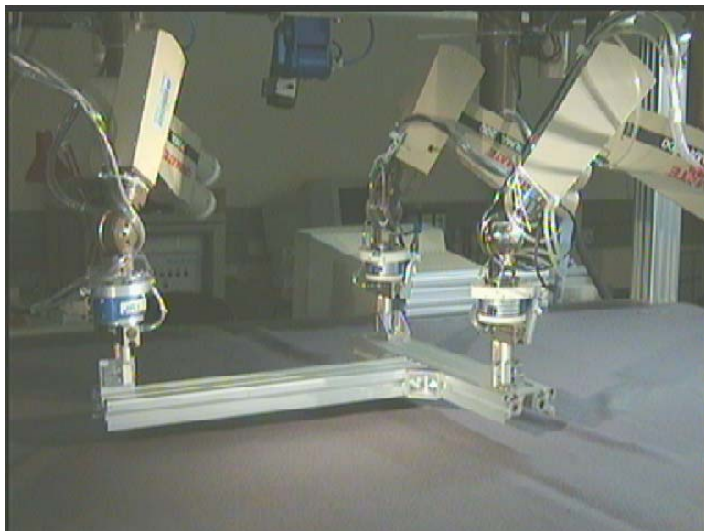


Figure 1: Three Puma-260 industry robots lifting a relatively long metal bar.

For a general-purpose arm/gripper system the following two types of uncertainties must be taken into account: grasping precision and slippage of the part in the parallel

jaw-gripper.

2.2 Independent Motions

The major approaches of coordinating the movements of several robots is a) the master-slave approach and b) the motion in a closed kinematic chain. In the master-slave approach forces are used to communicate between the master and the slave. This means that the inner forces are needed for calculating the new position.

In the closed chain approach the trajectory of all robots has to be calculated from one Computer within the desired cycle time. This approach does not correspond to the situation of several individuals coordinating their movements to perform a transportation task.

Therefore we are using an approach where all robots know the desired goal position independently. The robots are not synchronised, they run on several computers and they do not exchange data within cycle rate. Communication is only used to specify the trajectory.

The phases of a transportation task are:

- Every robot moves to the grasping position.
- After closing the hands the robots are going in some kind of compliance mode. The desired goal position is the actual position, and the inner forces have to be minimised.
- A new motion target is received, the motion parameters are calculated and the motion starts. Every manipulator starts at another slightly different moment. During this starting phase we have some kind of master-slave situation, the robots which are not already moving are still in compliance mode.
- When reaching the desired position the robots switch back to compliance mode.

Fig. 3(a) shows the effect of the master-slave mode. At the start and end of the motion the force in Y -direction increases. While two of the three robots are already on their way to the goal position the third one is still standing (slave mode). Because the speed is relatively high in comparison to Fig. 2 the robot is unable to compensate the forces properly.

2.3 Rapid Reinforcement Learning

A general unsupervised learning method for multiple-input–single-output system was presented in [11]. It is mainly characterised by the following features:

- Each controller output is defined by a set of fuzzy singletons (control vertices). The number of singletons equals the number of rules, their optimal value can be found by a learning method.
- Because the number of parameters that have to be changed in the learning process is relatively low, the convergence of a B-spline controller is quite fast.
- Changes to one singleton result in local changes of the control surface. This leads to a poor interpolation of untrained regions in the surface. Therefore the total input space has to be trained.

Assume that the Z -component of the translational part of one of the manipulators is to be controlled. To design this controller, the desired output for a given input vector is unknown. However, it can be assumed from Hooks-Law that the positional difference ($z_r - z_d$) is somehow proportional to ($F_{z,r} - F_{z,d}$), the difference between the real and desired force. If such a physical model is embedded in the learning process, the adaptation of control vertices which is normally a reinforcement learning problem, becomes much simpler.

To initialise the control vertices, the experts estimation values can be set if they are available. Otherwise, all control vertices are initialised with zero. In every control cycle¹ the output is added to the Z -component of the translational part of the corresponding transformation. By using the feedback information from the measured resulting forces the control vertices can be improved to get a better result in the next control cycle. Because $F_{z,r(t)}$ is a result of the controller's behaviour in the last control cycle (if there is no delay), the vertices of the last instead of the current cycle are modified. Therefore the input values of the last control cycle are used. If $F_{z,r(t)}$ is a result of several of the last cycles (the joint control takes several cycles), the modification has to be applied to the singletons of the last of these cycles.

3 Implementation Issues

3.1 Inputs/Outputs for Compliant Control

Assume that the robots should carry a rigid object together along a specified trajectory. For such a task we adopt the strategy that all robots are controlled. We design six controllers for each robot: translation and rotation along the tool - X , Y and Z axes. As input dimensions we selected x, f_x, dx, df_x which are the position, the related force, the speed and the force change, respectively. These inputs are modelled with ten uniform, triangular B-spline basis functions. Because the range of the sensor data is not *a priori* known, they are normalised to a range of $[-1, 1]$. The maximal values needed for the normalisation are adopted during the learning process.

3.2 Solving the Drift Problem

When using multiple force sensors an unpleasing drift effect can be observed, which is a result of the slightly different calibration. While sensor A detects no force, sensor B might measure a small value. The corresponding robot B will make a little position correction to reach the desired force, which exerts a small force on sensor A . If such an effect continues to exist, both robots may drift away from the desired trajectory.

One solution to overcome this problem is to limit the value of the integrated controller output. This means that the robots are allowed to leave the desired trajectory, but only in a certain small range that is just as much as needed for the internal force minimisation. For this purpose, we introduce another feedback to the cost function for learning, i.e. the drift from the desired trajectory. We suggest the following learning function:

$$\Delta d_{i_1, i_2, \dots, i_n}(t-1) = \epsilon (F_{z,r(t)} - F_{z,d} + (\gamma \cdot z(t)^3)) \cdot \prod_{j=1}^q N_{i_j}^{n_j}(x_j(t-1)) \text{ with } 0 < \epsilon \leq 1.$$

¹The control cycle for all robots is 20 ms in our experiment.

If the manipulator is near the desired trajectory, $z(t)$ is very small and the linear part ($F_{z,r(t)} - F_{z,d}$) dominates in modifying the control vertices. If the manipulator drifts away from the desired trajectory, $z(t)$ increases and the exponential part ($\gamma \cdot z(t)^3$) becomes more and more significant. γ can be used to adjust the effect of the exponential part for different drift allowances. The variables $z(t)$ as well as $x(t), y(t)$ are used as the fourth input of each controller respectively.

3.3 Repeated Practising

Generally, the learning process is performed in the following sequence:

1. Read the input values for the B-spline fuzzy controllers.
2. Calculate the controllers output and update the *SENSOR* transformation.
3. Store the input values $x_j(t)$ of the current step in a ring buffer.
4. Modify the control vertices.

This sequence is repeated every control cycle until a task is finished. The modified control vertices are used immediately in the next control cycle. The learning procedure for one complete task is called a *practice step*, which should be repeated several times so that for the same task the control vertices are adjusted optimally. Our experiments show that the learning rate ϵ directly influences the convergence speed. If ϵ is selected too small, the learning process takes a long time. If ϵ is too large, the learning procedure can cause oscillation. Our experience in selecting ϵ is that starting with an initial value, e.g. 0.01, ϵ is divided by two or more after a few *practice steps*.

4 Experimental Results

In the experiment three independent robots had to translate an object as can be seen in Fig. 1. The object weighs about 4kg, and exceeds the maximum lifting capacity of the manipulators. After two practice steps the forces on one sensor of either of the robots are depicted in Fig. 2. A four times faster motion results in the forces of Fig. 3.

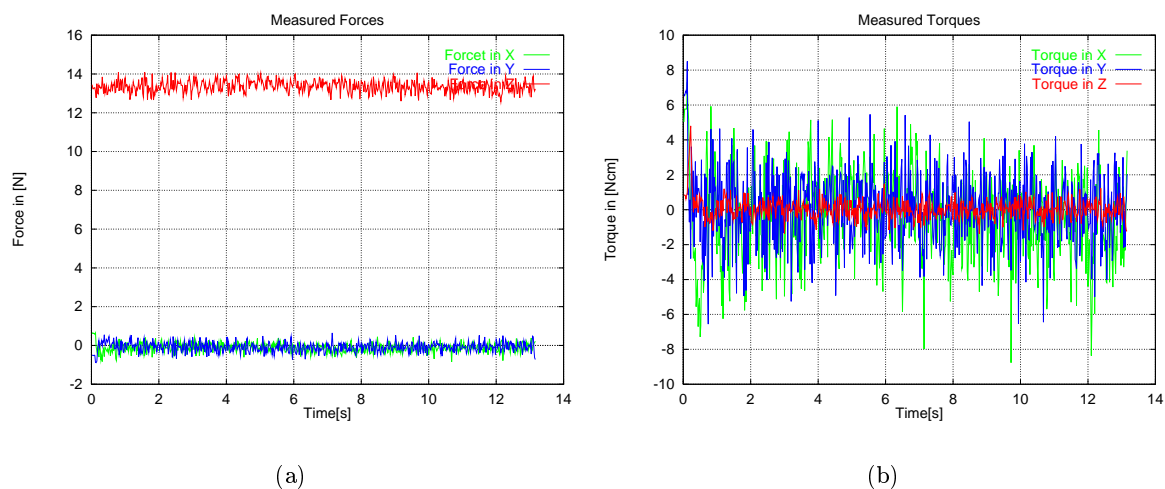


Figure 2: Forces (a) and torques (b) during a transportation task with three robots (speed: 2 cm/s).

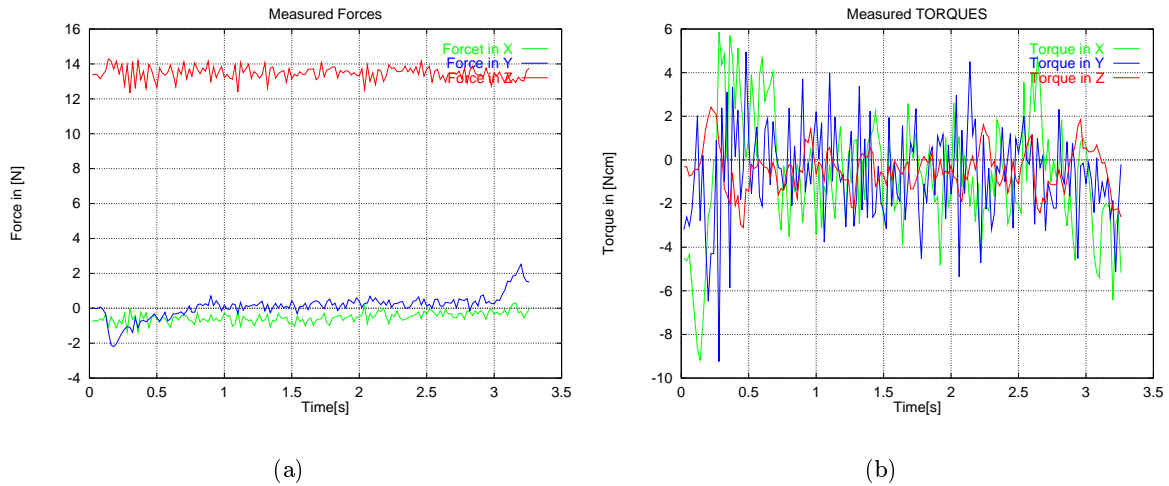


Figure 3: Forces (a) and torques (b) during a transportation task with three robots (speed: 8 cm/s). Note the forces in Y-direction are decreasing/increasing at the begin/end of the motion. This is caused by the master-slave behaviour of one of the three robots. If the translational speed is increasing the controller is unable to reduce the forces. This is made even worse by the drift compensation.

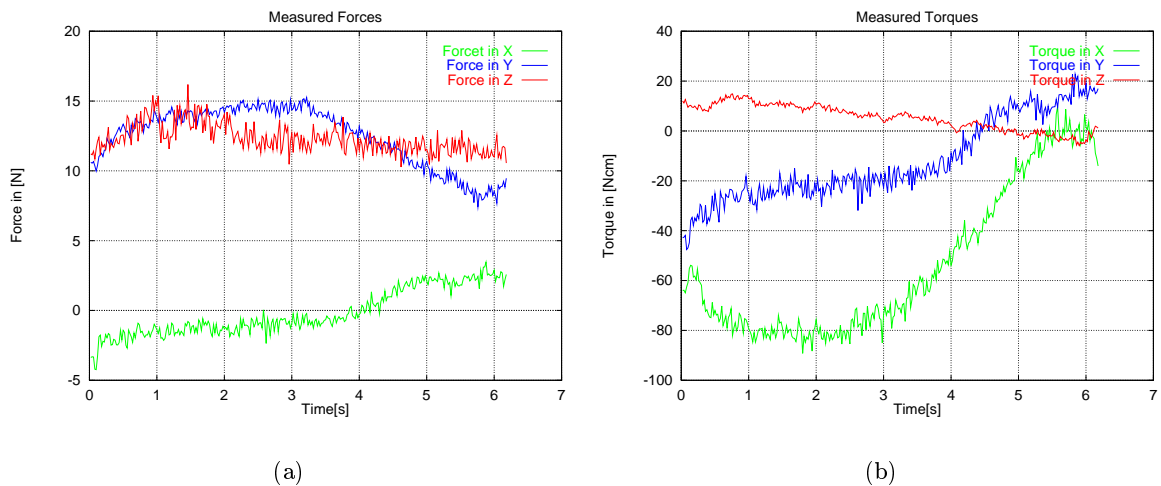


Figure 4: Forces (a) and torques (b) during transportation task without compensation (speed: 4 cm/s). It was impossible to perform a motion with more the 4 cm/s without any force compensation.

5 Exchange of Control Experience

Because B-spline fuzzy control is a very local learning method, only a small number of control vertices are trained (see Fig. 5), the major part of the singletons remain in their initial state, because no training examples are generated during the task. To extend the number of learned singletons the rule bases of the different robots are distributed. The learned singletons can be shared between the robots. Therefore robot A imports the singletons of robot B. The import strategy is fairly easy. Robot A steps through all of its untrained singletons and checks if one of the other robots has a trained value. If

this becomes true, robot *A* uses this value next time. That assumes that all controllers have the same value scope, and robots of the same type are used.

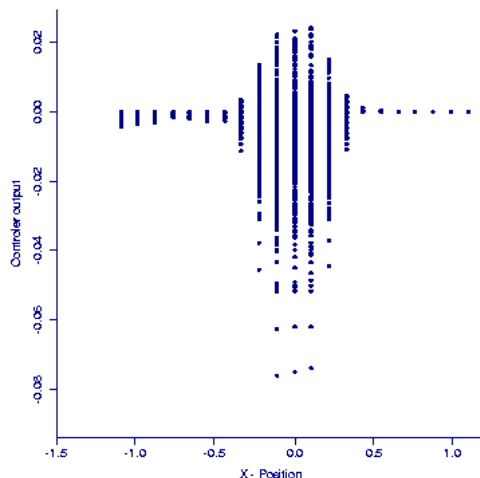


Figure 5: Projection of all values onto one position-axe of one of the controllers after two practice steps.

6 Conclusion

We have shown that hybrid force/position control such as multi-arm carrying can be solved with neuro fuzzy controllers. The B-spline model can be utilised for fusing different physical data. The approach is implemented on a multi-agent three-arm robot system using online-learning. The design of the controller is simple and the B-spline fuzzy controller can be trained in a straight-forward manner, because modification of the control vertices results in only local change of the control surface, and the training result can be shared between different robots.

References

- [1] O. M. Al-Jarrah and Y. F. Zheng. Intelligent compliant motion control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2610–2615, 1996.
- [2] J. S. Albus. A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Transactions of ASME, Journal of Dynamic Systems Measurement and Control*, 97:220–227, 1975.
- [3] C. G. Atkeson and J. C. Santamaria. A comparison of direct and model-based reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3557–3564, 1997.
- [4] M. Kaiser and R. Dillmann. Building elementary robot skills form human demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2700–2705, 1996.

- [5] K. Kiguchi and T. Fukuda. Fuzzy neural friction compensation method of robot manipulation during position/force control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 372–377, 1996.
- [6] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2282–2288, 1996.
- [7] M. H. Raibert and J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems*, 102:126–133, 1981.
- [8] B. R. Shetty and M. H. Ang Jr. Active compliance control of a puma 560 robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3720–3725, 1996.
- [9] J. Zhang and M. Ferch. Rapid on-line learning of compliant motion for two-arm coordination. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1853–1858, 1998.
- [10] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems*, 13(2/3):257–285, Feb./Mar. 1998.
- [11] J. Zhang, K. V. Le, and A. Knoll. Unsupervised learning of control spaces based on B-spline models. In *Proceedings of IEEE International Conference on Fuzzy Systems, Barcelona*, 1997.