Behavioural Modules for Force Control of Robot Manipulators

Williams, Tomos; Hardy, Nigel

Published in:
Robot Control 2000 (SYROCO'00): A Proceedings Volume from the 6th IFAC Symposium

Publication date:
2001

Citation for published version (APA):
Williams, T., & Hardy, N. (2001). Behavioural Modules for Force Control of Robot Manipulators. In *Robot Control 2000 (SYROCO'00): A Proceedings Volume from the 6th IFAC Symposium* (pp. 0-080). Elsevier.
http://hdl.handle.net/2160/31

BEHAVIOURAL MODULES IN FORCE CONTROL OF ROBOTIC MANIPULATORS

Tomos G. Williams, Nigel W. Hardy

Dept. of Computer Science, University of Wales, Aberystwyth
SY23 3DB, U.K. tgw@aber.ac.uk

Abstract: A behaviour-based architecture for the design of robot manipulator control in force mediated motions has been implemented. Its performance has been measured over a number of tasks to investigate the applicability of the approach. It has been found to offer a possible alternative to traditional control theory approaches in situations that do not lend themselves to an analytical approach. The proposed architecture offers a low cost means of implementing such control at an appropriate layer of abstraction for a non-specialist end user albeit at reduced accuracy and speed due to the limitations of the underlying external force control scheme. *Copyright© 2000 IFAC.*

Keywords: Robotic Manipulators, Robot Control, Architectures, Force Control, Flexible Automation

1. INTRODUCTION

Originally envisaged in the context of mobile robotics (Brooks, 1986; Arkin, 1989), behaviour-based methods have since been applied successfully to robot manipulator control (Stein and Paul, 1993; Malcolm, 1987; MacKenzie and Arkin, 1996). The work reported here investigates the benefits of behaviour-based controllers when applied to force mediated control.

We propose a behaviour-based architecture for the hybrid (force and position) control of a robot manipulator. The design of the proposed architecture, presented in §2, is based on the study of different behaviour-based approaches described in the literature and experience gained in a prototyping exercise (Williams, 1998; Williams and Hardy, 1998). The prototypes and the eventual architecture were implemented using the laboratory setup described below.

The experimental setup consisted of a Puma 560 robotic manipulator equipped with a six axis force sensor (BWC, 1986). An explicit force control scheme was developed (Dégoulange and Dauchez, 1994; Colbaugh and Engelmann, 1994) using the VAL II controller's external `ALTER` command receiving motion commands from the behaviour-based controllers executing on a Sun Sparc 5 workstation. There were inevitable limitations with this approach but the arrangement was suited to exploration of the potential capabilities of the behaviour-based architecture.

2. THE ARCHITECTURE

The selected architecture is shown in Figure 1. It is within this architecture that individual controllers which perform specific motions are built.

The `GetPositions` and `GetForces` behaviours provide the interface between the external sensors and the control architecture. They obtain the most recent robot position and tool-tip forces and make this data available to other behaviours. Similarly, the `MoveRobot` behaviour reads the motion commands requested by the other behaviours, packages them and issues them to the robot controller for execution. The use of interfacing modules through which sensor readings are obtained

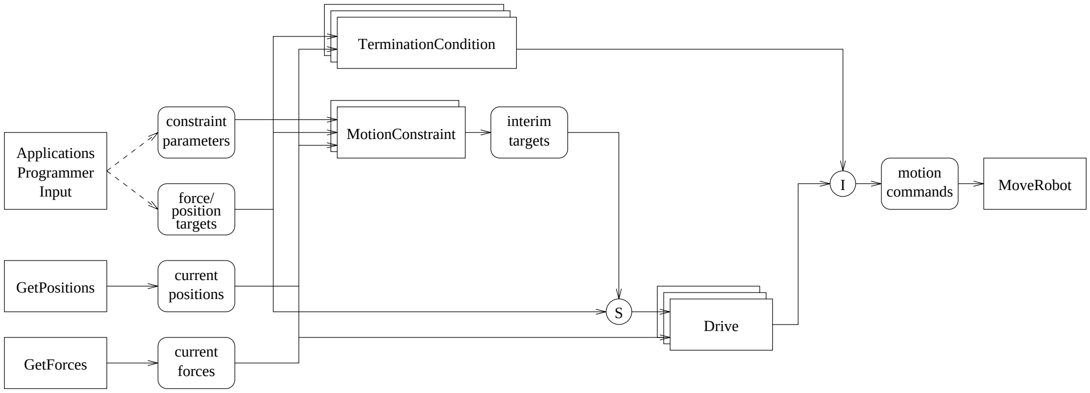


Fig. 1. The proposed architecture. Behaviours are represented as rectangles and communication channels as rounded boxes. The circles represent `inhibition` and `suppression` nodes where data from lower behaviours may be blocked, or replaced by data from higher level behaviours.

and actuator commands issued does not adhere to the strict behavioural doctrine that all behaviours should access to sensors and actuators directly (Brooks, 1986). However, successive prototype implementations demonstrated that this solution provided significant performance gains. They provide an interface to the underlying hardware which is optimal and constant for all the other behavioural modules. Previous implementations have also identified similar benefits (Stein and Paul, 1993; Alami *et al.*, 1998). In addition, commercial industrial robots are supplied with closed controllers which make it difficult and sometimes impossible for users to implement modules which directly interact with the robot actuators. Robot and controller manufacturers provide points of access at higher levels without jeopardising the integrity and reliability of their underlying real-time controllers (Rutlage, 2000).

Three `Drive` behaviours are present for any motion, one for each of the Cartesian translational axes. These can be of one of two types: position or force, a design for which is illustrated in Figure 2 which employs a primitive control strategy but serves to investigate the overall behavioural approach. Each drive behaviour is paired with a `Termination Condition` behaviour of the same type, position or force, working along the same axis. These inhibit the output from the respective drive behaviours if the error is within a tolerable range. Completion of a motion is signalled when all termination condition behaviours are active, indicating that all conditions are met. A study of common assembly operations recognised the need for a third type of termination condition behaviour, that of `Edge`. Paired with a force driving behaviour it indicates that the edge of a surface has been crossed; this is detected by travel, without contact, for a specified distance along the axis.

`MotionConstraint` behaviours are the means by which relationships between controlled axes are imposed. Two such constraints were implemented. The `StraightLineMC` behaviour imposes positional constraints by calculating a number of via points on the straight line path between the start and target position. These points suppress the position targets and provide a number of interim targets along the required path. `StraightLineMC` updates these targets as the motion progresses. The second motion constraint is the `ForceMC` behaviour which limits frictional forces and ensures that contact is maintained for surface following motions. It operates by disabling translational motions unless the reaction force along the relevant axis is within a specified range.

3. EXPERIMENTATION AND RESULTS

The aims of the experimentation stage were to validate and assess the performance of the behaviour-based architecture shown in §2. This was achieved by implementing and testing behavioural configurations to perform a range of force mediated tasks. Each configuration was constructed within the structure of the behaviour-based architecture with the correct combination of drive, termination condition and, if required, motion constraint behaviours. Four behavioural configurations are described.

3.1 *Guarded Move*

For this task the manipulator, starting in free space, was required to move along one of its tool axes until contact with a surface was detected; contact was then maintained, as specified in Singh and Popa (1995). The `GuardedMove` controller, shown in Figure 3 illustrates how specific controllers were constructed within the behaviour-based architecture. In addition to the required interface modules `GetPositions`, `GetForces` and `MoveRobot` the controller consisted of a force

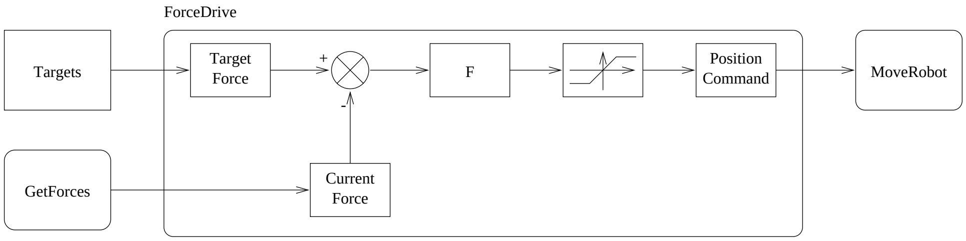


Fig. 2. An example of a Force Drive behaviour. The error between the current and target force along its respective axis is calculated and, depending on the sign of that error, the behaviour instructs the robot to move in the direction which would reduce that error along that axis

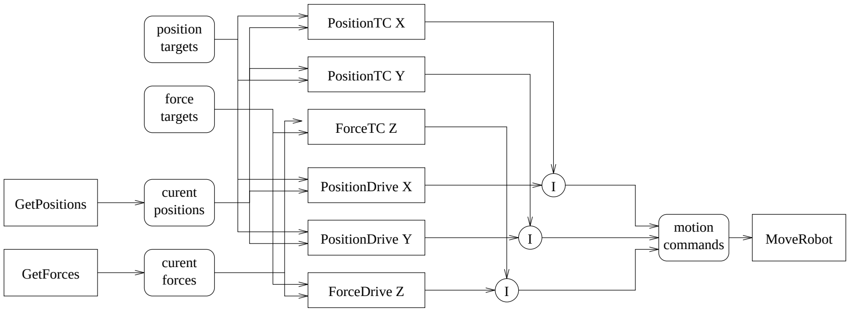


Fig. 3. Controller components and connections for the guarded move task.

driving and terminating behaviour for the translational Z axis: `Force DriveZ`, `ForceTCZ`. The `ForceDriveZ` behaviour provided the motion commands along the Z axis and instructed the manipulator to move towards or away depending on the force value. The `ForceTCZ` behaviour reduced hunting around the desired reaction force by inhibiting the motion commands from the `ForceDriveZ` behaviour when the error between the sensed and target force fell within a tolerable range.

The robot, holding a felt tipped pen of unknown stiffness in its gripper, was required to descend onto paper secured to a hard wooden surface. The controller performed the task satisfactorily for all of the test runs moving at, on average, a tool tip velocity of $0.76mm/s$. A force tolerance of ±0.25 N was specified. Figure 4 shows the results for tests where the controller was instructed to lower the robot with different force termination conditions along the Z axis. This first version of `ForceDriveZ` simply instructed the manipulator to move forward the minimum possible step for each command sample until the reaction force was detected. If this force exceeded a certain threshold, the manipulator would retract the tool. Due to the simplicity of the algorithms employed in this behaviour the controller was unsurprisingly unable to compensate for the time delay in the system. This results in unstable control with the manipulator oscillating around the zero error mark. A number of different control laws were tested within the `ForceDrive` behaviour, the most successful of which employed a damped control law, matching the delay inherent in the control loop. The results for this augmented controller are also shown in Figure 4. This example illustrates the ease with which amendments could be made, in this case by choosing a different `ForceDrive` behaviour, to improve the controller's performance. This is due to the inherent modularity and flexibility of the architecture which specifies the interface to the behaviours, allowing for alternative modules to be easily developed and substituted into a controller without causing major repercussions.

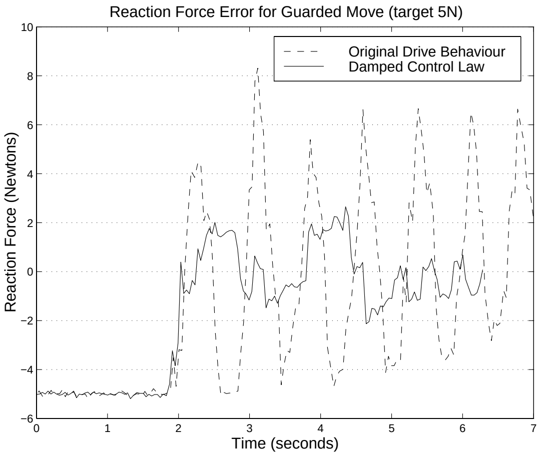


Fig. 4. The Z force during a guarded move using a simple behaviour and one with a damped control law.

3.2 *Surface Following*

This task (De Schutter and Van Brussel, 1988; Whitney, 1987; Demey *et al.*, 1997; Whitcomb *et al.*, 1997) required the manipulator to maintain

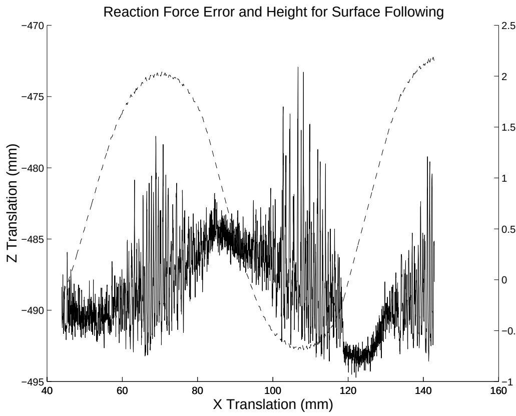


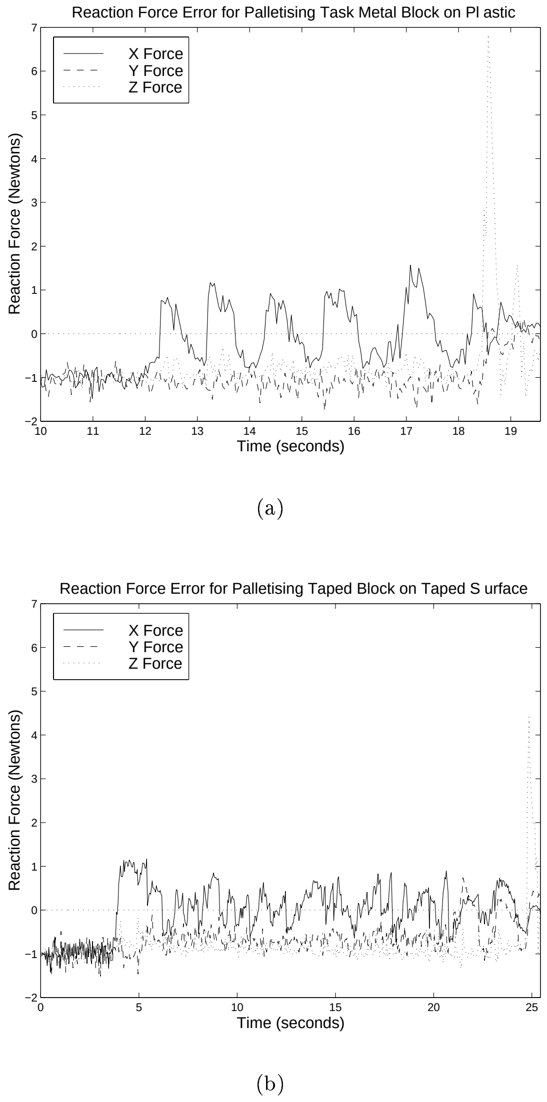
Fig. 5. The contour and force error for the `DrawStraightLine` controller writing on the corrugated roofing plastic.

a contact force along its Z axis while moving along the surface to a target in its XY plane. The configuration of the `DrawStraightLine` controller was very similar to that of the `GuardedMove`. Force along the Z axis was maintained and regulated by the `ForceDriveZ` and `ForceTCZ` behaviours and translational motion was achieved by providing target positions to the X and Y position drive and termination condition behaviours. The only amendment required to the `GuardedMove` controller shown in Figure 3 was the implementation and integration of the `StraightLineMC` which imposed the straight line relationship between the X and Y position controlled axes by injecting interim position targets to the drive behaviours.

The `DrawStraightLine` controller proved capable of performing the surface following task under a wide range of conditions. It was initially tested on a flat, hard surface set at different angles to the tool Z axis. The controller was then required to write on 20mm corrugated roofing plastic. Figure 5 shows the contours of the corrugated sheeting taken from results of the trajectory of the tool tip during execution of the draw line task and the reaction forces demonstrate that contact was maintained between the pen and surface resulting in a continuous line being drawn. The phase difference between the vertical position and reaction force error is due to the maximum force error occurring at the steepest part of the surface.

3.3 *Palletising*

The block-in-corner or palletising task (Whitney, 1987; De Schutter and Van Brussel, 1988) requires the manipulator to maneuver a gripped steel block from free space into a corner, colliding and maintaining contact with each of the three sides one at a time until the specified reaction forces are detected simultaneously along all three translational degrees of freedom. The design of the `Palletise` controller consisted of a force driving behaviour along each axis. These were provided with target forces, the signs of which indicate the quadrant in Cartesian space in which the corner is located. Each force driving behaviour was paired with a force terminating condition behaviour which regulated and maintained contact along each axis as and when collision occurred until all three termination conditions were met indicating successful part mating.



(a)

(b)

Fig. 6. Results for the `Palletise` controller tested on different surfaced materials with a target reaction force of 1 N. (a) Metal block on plastic; (b) Taped block on tape.

The `Palletise` controller was tested with objects of different surfaced materials with coefficients of friction ranging from 0.21 (metal block on plastic surface) to a challenging 0.57 (adhesive insulation tape covered block on adhesive insulation tape covered surface). Results from successive runs indicated that the controller could perform the task

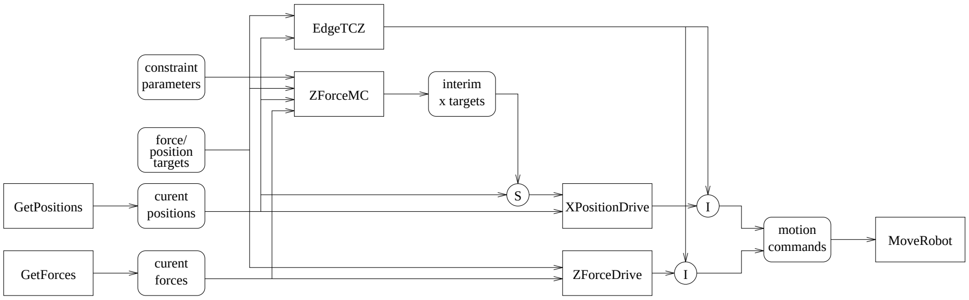


Fig. 7. Controller design for the chamfer crossing stage of a peg in hole task.

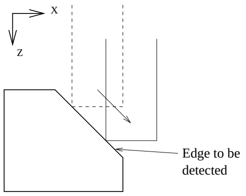


Fig. 8. The chamfer crossing assembly task.

successfully under a range of task conditions and requirements. Figure 6 shows the results for two such runs using (a) a metal block on a plastic surface and (b) a taped block on a taped surface both with a target reaction force of 1N specified along each axis. These results illustrate how progress is made starting in free space where no reaction forces are detected along any of the axes then, collision occurs and contact is maintained along each axis in turn until the corner is found. This task was successfully performed under a range of task conditions.

3.4 *Chamfer Crossing*

Chamfer crossing, the first stage of the classic compliant motion control application peg-in-hole (Mason, 1981; Raibert and Craig, 1981; Sailsbury and Craig, 1982), was the final task to be tackled. This motion is shown in Figure 8 and tests a controller's ability to maintain contact with an angled surface and to detect the edge of the surface which, in this case, coincides with the opening of the hole.

The design of the `ChamferCrossing` controller is shown in Figure 7. The `PositionDriveX` behaviour is provided with a position target beyond the hole opening and, therefore, continually provides motion towards the hole. The `ForceDriveZ` aims to maintain contact between the peg and the chamfer surface but, since it is set at an angle of 45° it requires the assistance of the force motion constraint behaviour, `ZForceMC`, which detects loss of contact and suspends motion along the X axis by supplying target positions which are equal to the current positions to the `PositionDriveX` behaviour. This allows the `ForceDriveZ` behaviour to re-establish contact with the surface. The termination condition for this task is determined solely by the edge termination condition behaviour along Z axis `EdgeTCZ`. It detects the edge by storing the recent history of the tool X and Z position and, if it detects vertical downwards motion, deems that the edge of a surface perpendicular to the Z axis has been reached and the motion is terminated.

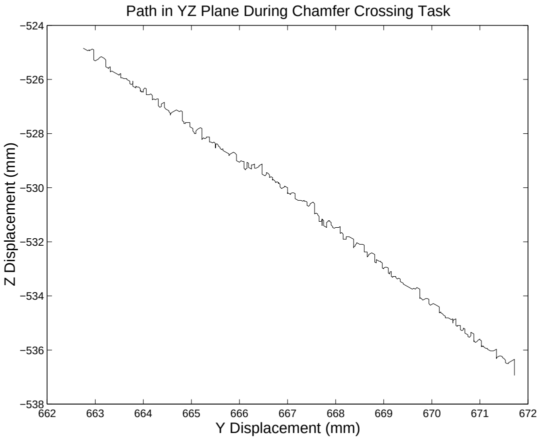


Fig. 9. Results for the Chamfer Crossing controller showing the path followed by the tool along the chamfer set at 45° to the Y and Z axis.

In order to test the robustness of the `Chamfer Crossing` controller the task was performed using different surfaced materials providing coefficients of friction ranging from 0.21 to 0.57. Again, the behavioural controller performed the task consistently successfully for different task configurations. Results from one such run, shown in Figure 9, demonstrate the path followed along the chamfer ending in a vertical motion indicating that the edges of the surface and hence the opening of the hole has been reached.

4. CONCLUSIONS

The flexibility of the architecture was demonstrated by the ease with which modules employing alternative control algorithms for the guarded motion task could be substituted for the original force driving behaviours, enhancing the controller's performance without having to alter any of the other modules. The use of motion constraining behaviours which provided additional capabilities and which were accommodated without the need to alter other modules illustrated the extendibility of the architecture.

The lack of an explicit environment model led to solutions not tied to a model of a particular working environment. This was demonstrated by identical controllers being able to perform adequately under a range of task conditions using tools of different stiffness and coefficients of friction. The representation of the task employed by the behaviour-based architecture was user oriented. Motions were divided according to intuitive, externally observable tasks: “Drive robot along the X axis”, “impose a straight line relationship between the X and Y position controlled axes”. These behavioural modules provided an appropriate layer of abstraction for a non-specialist end user.

The work presented in this paper has demonstrated the viability and possible benefits of a behaviour-based approach for force control of robot manipulators. It has also revealed a number of areas where this work could be extended to investigate some of these benefits further. This would of course be assisted by improved communication with the robot in contrast with the environment used.

5. REFERENCES

Alami, R, R Chatila, S Fleury, M Gallab and F Ingrand (1998). An architecture for autonomy. *International Journal of Robotics Research* **17**(4), 315–337.

Arkin, Ronald C (1989). Motor schema-based mobile robot navigation. *International Journal of Robotics Research* **8**(4), 92–112.

Brooks, Rodney A (1986). A robust layered control-system for a mobile robot. *IEEE Journal of Robotics and Automation* **2**(1), 14–23.

BWC (1986). *FS6-120A 6-axis Force Sensor Software Manual.* Barry Wright Corporation, Watertown, Massachusetts.

Colbaugh, R and A Engelmann (1994). Adaptive compliant motion control of manipulators: Theory and experiments. In: *International Conference on Robotics and Automation.* IEEE. San Diego, California. pp. 2719–2726.

De Schutter, J and H Van Brussel (1988). Compliant robot motion 2. A control approach based on external control loops. *International Journal of Robotics Research* **7**(4), 18–33.

Dégoulange, E. and P. Dauchez (1994). External force control of an industrial Puma 560 robot. *Journal of Robotic Systems* **11**(6), 523–540.

Demey, S., H. Bruyninckx and J. De Schutter (1997). Model-based planar contour following in the presence of pose and model errors. *International Journal of Robotics Research* **16**(6), 840–858.

MacKenzie, D. and Ronald C. Arkin (1996). Behavior-based mobile manipulation for drum sampling. In: *Proc. 1996 Int. Conf. on Robotics and Automation.* Minneapolis, MN. pp. 2389–2395.

Malcolm, Chris (1987). Planning and performing the robotic assembly of soma cube constructions. Master’s thesis. Depatment of Artificial Intelligence, University of Edinburgh.

Mason, M. T. (1981). Compliance and force control for computer-controlled manipulators. *IEEE Transactions on Systems Man and Cybernetics* **11**(6), 418–432.

Raibert, M. H. and J. J. Craig (1981). Hybrid position-force control of manipulators. *Transactions of the ASME — Journal of Dynamic Systems Measurement And Control* **102**(2), 126–133.

Rutlage, Gary J (2000). The PC and its influence on robot controllers. In: *ICRA, IEEE International Conference on Robotics and Automation.* Vol. 1. San Francisco, California. pp. 717–721.

Sailsbury, J K and J J Craig (1982). Articulated hands: force control and kinematic issues. *International Journal of Robotics Research* **1**(1), 4–17.

Singh, S K and D O Popa (1995). An analysis of some fundamental problems in adaptive control of force and impedance: behaviour, theory and experiments. *IEEE Transactions on Robotics and Automation* **11**(6), 912–921.

Stein, Matthew R and Richard P Paul (1993). Behavior based control in time delayed teleoperation. In: *International Conference on Advanced Robotics.* Tokyo, Japan. pp. 223–227.

Whitcomb, L. L., S. Arimoto, T. Naniwa and F. Ozaki (1997). Adaptive model-based hybrid control of geometrically constrained robot arms. *IEEE Transactions on Robotics and Automation* **13**(1), 105–116.

Whitney, Daniel E. (1987). Historical-perspective and state of the art in robot force control. *International Journal of Robotics Research* **6**(1), 3–14.

Williams, Tomos G (1998). On the Use of Behavioural Modules for Force Control of Robot Manipulators. PhD thesis. University College of Wales, Aberystwyth.

Williams, Tomos G and Nigel W Hardy (1998). A behaviour-based architecture for force control of robot manipulators. In: *IEEE International Conference on Systems, Man and Cybernetics.* San Diego, California. pp. 3532–3537.