Experiments in Reflex Control for Industrial Manipulators
Wyatt S. Newman* and Michael S. Branicky
Dept. of Electrical Engineering and Applied Physics
and
Center for Automation and Intelligent Systems Research
Case Western Reserve University
Cleveland, Ohio 44106

Abstract
Experiments in automatic collision avoidance for robots using acceleration-based reflex control are described. Acceleration-based reflex control responds to acceleration commands from higher levels, as opposed to prior reflex control approaches based on position commands from higher levels. Reflex control is exerted in configuration space and assumes either inherent dynamic decoupling of the robot links or feedback decoupling. Implementation of acceleration-based reflex control is described for 3-D collision avoidance using a General Electric GP132 robot. Collision avoidance with respect to stationary obstacles is guaranteed. In addition, performance measurements illustrate how the reflexes introduce no significant distortion to higher-level controllers when reflex action is not required.

1 Introduction
Real-time obstacle avoidance for high-speed robots can be achieved via a technique called reflex control [1, 2, 3, 4]. The reflex control layer can be introduced in either a serial or parallel manner. In a parallel implementation, the reflexes monitor the state of the robot (positions and velocities) and the state of the world (represented in configuration space) and "step in" whenever it is necessary to specify commands that will ensure continued safe operation. That is, the reflexes are always present but intervene only when it is necessary to prevent imminent danger. It is this type of operation that prompts the analogy with biological reflexes.

Reflex control can be implemented in series with the same result: it can receive requests from higher layers and either deny or approve them based on whether a collision is imminent or not. This is the implementation we adopt. Conceptually, reflex control is consistent with layered, hierarchical control systems [5, 6]. In a hierarchical control structure, the reflex control layer is inserted at a layer below the level of abstraction of path planning. Position-based reflex control may be incorporated at a level above servo control; alternatively, acceleration-based reflexes (considered here) are installed below the servo control level.

Reflexes are merely a limited form of intelligence which prevents the robot from exceeding joint limits and from striking itself or objects in its environment. They are low in the control hierarchy, but that does not limit their effects. Just as motor saturation might prevent a linear control law from producing the torques it requests, so do the reflexes intervene to deny any requests from higher levels which would result in collision. And just as the effect of saturation is transparent to torque requests which do not exceed the saturation level, so too should the reflexes be transparent when requests are not dangerous.

In [3, 4], artificial reflexes for automatic obstacle avoidance were proposed and tested on two manipulators. In [5], reflex control was demonstrated on a high-speed 2-link planar arm. In [4], implementation of reflexes in 3-D on a General Electric GP-132 industrial robot is described. In both of these works, as in the present paper, reflex control depended on decoupled joint dynamics, and was exerted in joint space. Reflex control in joint space assumes the existence of a map of reachable obstacles as represented in Configuration Space [7]. These assumptions may seem restrictive, since joint dynamics are generally considered highly coupled and Configuration Space obstacle computations are considered time consuming. In [3], joint dynamics are decoupled by design, a technique further described in [8]. In [4], an industrial robot was used in reflex experiments. Although no joint decoupling was intentionally designed in, the joints were found to be nearly dynamically decoupled due to the dominance of motor inertia reflected through the high-ratio harmonic drives [9].

Since industrial robot designs typically employ high-ratio transmissions, joint dynamics commonly exhibit only weak dynamic coupling. Minor joint coupling can be rejected by feedback linearization [10]. In contrast, feedback linearization for decoupling of hand dynamics in Cartesian coordinates or "Operational Space" [11] requires gross compensation, thus limiting the valid speeds and accelerations to very low values.

In [2], automatic obstacle avoidance was incorporated with joint control in a single algorithm. In the experiments of [3] and initial work of [4], reflexes were separated out from direct joint control and implemented in a hierarchy in which the reflex control layer resided between a planning layer and a servo control layer. In this organization, the planner delivered requests for position setpoints, requests were considered by the reflexes for approval or substitution, and the resulting position commands were executed at the servo level.

In [1], a different hierarchical organization was proposed. In this work, reflexive obstacle avoidance is described with respect to requested joint accelerations, as opposed to position requests. Reflexes thus enter the hierarchy below the servo control layer. Such an organization permits use of arbitrary joint torque-based control algorithms, i.e., for adaptation, logical control mode switching, or risky experimentation. It is the responsibility of the reflexes to execute rapidly any requested vector of joint accelerations, unless
the requested accelerations would lead to an inevitable collision. Such a controller was proposed, in theory, in [1]. This paper presents experimental evidence of the feasibility of that approach.

2 Direct Acceleration Control Emulation

Reflex control using direct acceleration control emulation assumes the existence of some acceleration-based controller at a higher level. The reflexes are responsible for preventing collisions, yet they should normally be transparent to higher-level control.

In direct acceleration control emulation our system looks like Fig. 1. The servo controller accepts position and velocity commands from the reflex module, and issues an acceleration command to a linear (or feedback linearized) plant. Independent (or feedback decoupled) second-order acceleration commands are "encoded" in terms of an anticipatory setpoint to a simple proportional plus derivative controller. Accelerations produced by the PD controller conform to:

\[ \alpha_{\text{command}} = K_p (z_{\text{setpt}} - z_{\text{actual}}) + K_I (\dot{z}_{\text{setpt}} - \dot{z}_{\text{actual}}) \]  

Fig. 1: Reflex Controller, Direct Torque Control Emulation

The reflex controller invents a virtual setpoint which has the result of converting acceleration requests into acceleration commands to the linearized, decoupled plant. Specifically, acceleration commands will track acceleration requests if the reflex controller chooses

\[ z_{\text{setpt}} = z_{\text{actual}} + \frac{K_p}{K_p} \dot{z}_{\text{actual}} + \alpha_{\text{req}} \]  
\[ \dot{z}_{\text{setpt}} = 0 \]

The desired acceleration, \( \alpha_{\text{req}} \), is executed if the PD controller is commanded to head towards the point \( z_{\text{setpt}} \), at least until the system state changes significantly. If the virtual setpoints are updated much faster than the bandwidth of the PD controller, then the corresponding desired accelerations are accurately realized. This transformation of acceleration commands into equivalent instantaneous position commands permits a geometric specification of the reflex algorithm. That is, an acceleration request is granted by approving the corresponding setpoint, and this setpoint is approved if it would not result in a collision under PD control. The reflexes exert their influence by refusing to update new virtual setpoints, which would lead to system convergence to the last approved setpoint. Invoking the velocity and/or PD controller constraints specified in [1], each approved setpoint is guaranteed to be attainable by the system without overshoot.

Having converted acceleration requests into equivalent candidate virtual setpoints, the approval of acceleration requests reduces to inspection of configuration space in the region between a current pose and the candidate virtual setpoint. Having transformed the acceleration approval process into a search in configuration space, we can invoke the search algorithms for position-based reflex control tested in [3, 4].

If the reflex controller is fast enough, the reflex and servo modules will cancel each other's dynamics, resulting in a zero net influence. The success of such dynamics cancellation or reflex "transparency" is a key condition of effective reflex control.

3 Reflex Control in Higher Dimensions

Our description of reflex control to this point has been limited to one dimension. In only one dimension, the search for potential collisions in discretized configuration-space consists of examining all C-Space elements in line between the current position and the goal position.

Reflex control in one dimension is generalizable to higher dimensions. If a decoupled plant is assumed, then each second order system may be treated like the one-dimensional system described above. The difference with higher dimensions is that the regions which must be inspected for obstacle avoidance are no longer simple distances, but areas, volumes, or hypervolumes whose vertices are given by \( z_{\text{actual}} \), to \( z_{\text{setpt}} \), (or \( z_{\text{max}} \), in each dimension \( t = 1, \ldots, d \).

Another difference with higher dimensions is the effect of geometric coupling among the subspaces [3, 1]. This effect is shown in Fig. 2. In order to avoid choosing the overly conservative setpoint, some choice must be made as to which axis gets precedence. This choice may be made based on distance to the goal point or some other optimizing criterion. In the present implementation, however, a simple list of joint precedences is used to eliminate such conflicts.

4 Computational Considerations

Our two major requirements of the reflex controller are guaranteed obstacle avoidance and transparent operation in the midst of no obstacles. The second criterion is harder to achieve, especially if transparency is required for maximum velocity in all degrees of freedom.

In general, the reflex controller must search a \( d \)-dimensional rectangular prism with sides given by \( \Delta x_{\text{max}}(t = 1, 2, \ldots d) \) to determine if candidate virtual setpoints are safe or not. This prism contains all possible
trajectories resulting from presumed sudden application of maximum controlled braking. We refer to this bounded region as the "arrest region". The use of boundary searching can substantially reduce the cycle time of the reflex controller relative to exhaustive searches of the arrest region.

In a naive implementation of the reflex controller an exhaustive search is performed of the arrest region on each reflex control cycle. For example, assume a discretized d-dimensional C-Space map and a search distance to the virtual setpoint of n bins (configuration space discretizations).

Then, with an exhaustive search, on each iteration we must search an $n^d$ array of bins. This results in safe operation and may be effective for small $n^d$. However, if $n$ is large the time it takes to search $n^d$ bins may result in non-transparent reflexes (i.e., the robot slows down because the reflexes are not updating setpoints quickly enough).

To alleviate the problem, the speed of the reflexes can be increased by performing boundary rather than exhaustive searches. Referring to the example above, this means that if the robot has moved m bins along each joint since the last update, only the $n^d - m^d$ boundary bins—not the exhaustive $n^d$ bins—need be searched to ensure safety.

For higher dimensions, even boundary searching may pose a problem. For example, using $d=4$, $n=10$, and $m=9$ in the above calculations leaves 3439 bins to search. If $n=6$, 465595 bins must be searched, and this search must be completed "fast" with respect to the desired bandwidth of direct acceleration control emulation. For example, assuming a bin inspection rate equal to that of our current implementation, (bin inspection rate $\approx 117$ kHz), reflex iteration rates in four dimensions would be about 34Hz, and extension to 6-D would reduce the reflex update rate to 0.25 Hz. For higher dimensions, discretizations have to be chosen such that between updates the robot moves at most one bin in a subset of its degrees of freedom at maximum-speed in all joints. Parallel processing could also be used. However, unless data reduction (run-length coding or $2^n$ trees) is utilized, storage constraints will outstrip the reflexes in processing requirements because storage and exhaustive searching grow as $n^d$ while boundary searching grows $O(n^{d-1})$ (where $n >> d$).

In our experimental 3-D implementation on the G.E. GP132 robot, configuration space was discretized as a $128 \times 128 \times 128$ array, corresponding to joint angular displacements of $2.34^\circ$, $0.86^\circ$, and $0.94^\circ$ for joints 1, 2, and 3, respectively. At this level of discretization, the maximum required search volume was nearly 20000 bins, corresponding to about 6Hz reflex update rate. At only 6Hz update rate, the reflexes did introduce significant distortion of the commands from higher control levels, exhibited as a speed limit on the joints during fast, coordinated-joint moves. Upon implementing the boundary search technique, reflex computation speeds were increased from about 6 Hz to 400 Hz. The increased speed afforded by boundary searching resulted in a nearly transparent reflex controller for the first three joints of the GP132 robot.

5 Reflexes in the Control Loop

If reflex control is to be a part of real-time control loops, the delay time of collision-detection computation must be considered. If no obstacles are present, requests are approved after each cycle (frequency $f_c = 1/t_c$) and the reflexes may be modeled as a zero-order hold and delay as shown in Fig. 3. The impulse response of this reflex control module is

$$g(t) = \begin{cases} 
0, & |t| < t_r \\
1, & t_r \leq |t| < 2t_r
\end{cases} (4)$$

with corresponding Fourier transform

$$G(\omega) = \frac{\sin(\omega t_r/2)}{\omega} e^{-\omega t_r/2} (5)$$

which may be solved numerically for the 3dB cutoff frequency,

$$\omega_c \approx \frac{2}{t_r} (6)$$

In our experiments, the worst-case update time of the reflexes was 400Hz. For a sampling rate of 400Hz, the 3dB frequency of the reflex module is about 180Hz. For comparison, the maximum stable bandwidth of a linear PD controller for the GP132 was found empirically to be about 2.5 Hz. Thus, the extra dynamics introduced by the reflex controller is unimportant relative to lower frequency unmodelled dynamics of the robot (such as mechanical resonances).

Equations (3), (4) describe the transfer function associated with the reflexes' sampling acceleration requests and inspecting configuration space to determine the feasibility of approving the requests. Assuming an acceleration request is approved, it is still not issued as a direct command. Instead, the acceleration command is encoded as an equivalent setpoint to a PD controller, as specified in equation 1. Upon initial input of a new setpoint, the PD controller acts to control the torque (and thus acceleration) which has been approved by the reflexes. Between updates of the virtual setpoints, however, the PD controller causes the acceleration command to decay.

We next consider the dynamics introduced by the PD controller. Let us define $K_p$ and $K_v$ in terms of $\omega_n$ for critical damping, i.e., $K_p = \omega_n^2$, $K_v = 2\omega_n$. Using $\ddot{z} = z_d - z_{actual}$ (and similar notation for $\dot{z}$ and $\dot{\ddot{z}}$), we can rewrite (1) as

$$\begin{align} 
\delta + 2\omega_n \dot{\ddot{z}} + \omega_n^2 \ddot{z} &= 0 \tag{7}
\end{align}$$

In our implementation, the $z_d$ is held constant over the sampling time of the reflexes. The desired position and velocity are zero, so $\dot{\ddot{z}} = -v$ and $\dot{z} = -\alpha$. The effect of holding $z_d$ constant over a time step is given as:

$$\ddot{z}(t) = \dot{z}(0)e^{-\omega_n t} + \left[(\dot{z}(0) + \omega_n \ddot{z}(0))t\right] e^{-\omega_n t} (8)$$

where $t$ is reset to zero at each update of the reflexes.
(8) can be differentiated twice to show that acceleration requests presented to the robot decay as
\[
\alpha(t) = \alpha_d \left[ 1 - (\omega_n t)^2 \left( 1 + \frac{\omega_n^2}{\omega_d^2} \right) \right] e^{-\omega_n t}
\]
where, again, \( t \) is reset to zero at each update of the reflexes. In our experiments, 3-D reflex sampling rates were 400Hz, and the corresponding \( \alpha(t)/\alpha_d \) was never less (theoretical worst case) than 0.96.

6 Performance

Three-dimensional reflex control was implemented on an industrial robot: the General Electric GP132. Fast, 3-D C-Space transformation algorithms for this robot are described in [4]. Control system retrofit and measured parameters of this machine may be found in [9]. Torque control of the individual joints is executed by the CAISR Multiprocessor System [12], which incorporates 20MHz MC68020-based VME-compatible single-board computers, analog and digital I/O, shared memory and a Sun Workstation host.

Effective execution of reflex control depends on rapid inspection of Configuration Space. In our representation, Configuration Space is discretized into a regular array of "bins". Details of the configuration-space representation may be found in [15]. The bin-inspection algorithms searched configuration-space poses at a rate of 117kHz. With boundary searching, this resulted in cycle times of about 400Hz. "Transparency" of the reflexes, i.e. introduction of negligible dynamics under safe conditions, was tested and is reported here.

The data presented here was obtained during actual moves of the GP132 robot. Samples were saved in a long record in RAM with sampling at approximately 1kHz. Noise on the torque plots is due to control gains amplifying tachometer noise.

The moves shown correspond to a demanding tracking test. The GP132's inverse kinematics and Jacobian were used to command position and velocity profiles which corresponded to sinusoidal motion of the end-effector in a plane of Cartesian space. Such hand motion required coordination of three links. In [9], this same test trajectory was used to measure the accuracy of high-performance computed torque control for precise tracking. Thus, the test conditions are especially demanding on the reflexes to approve torques rapidly and precisely for effective tracking.

Measured tracking performance with and without reflexes is shown in figures 4 and 5, respectively. The chosen trajectory is cyclic at 1Hz. For the chosen trajectory, each joint is driven into torque saturation over some part of the cycle. At saturation torque (i.e., saturation acceleration), distance to the virtual setpoint, and thus the required inspection region of C-Space, is at a maximum. Thus, the trajectory tests worst-case cycle times of the reflex controller. Even under these worst-case conditions, the acceleration requests are approved at high enough rates that reflex action is seemingly transparent.

Figures 4 and 5 show that our reflex controller does not disturb the normal operation of a higher-level servo controller. This is in spite of the fact that each acceleration command from the servo controller must be actively evaluated by the reflexes before it is approved for execution.

In the test results of Figs. 4, 5, joint angle, angular velocity and joint torque are plotted together vs time for each of the three joints. In these plots, time is in seconds, \( \theta \) is in radians, \( \omega \) is in radians per second, and \( \tau \) is normalized to \( \tau/\tau_{\text{max}} \). (For joint 1, \( \theta_1 \) is off the vertical scale).

It is clear from the data that the reflex controller did meet the "transparency" criterion. What is not immediately obvious from the tracking data is that the reflex controller is effective in preventing motion into an obstacle. In fact, construction of the reflex controller ensures that the robot is incapable of striking any known, static obstacle. Inspection rates in the presence of obstacles are even faster than the 400Hz rates obtained in free space.

Although we present results using a tracking controller, the nature of the higher-level control algorithm is irrelevant. In fact, acceleration requests corresponding to bang-bang control, pure noise, d.c. steps, or any unintentional or even deliberately malicious inputs are all treated the same way, and safety is guaranteed in each case.

7 Reflexes in the Control Hierarchy

The reflexes have been implemented as a layer in a hierarchical control system. They have operated successfully in both the earlier position-based reflex control approach as well as the more recent acceleration-based approach described here. They have operated in conjunction with PD, bang-bang, sliding-mode and computed-torque controllers. Higher-level commands have been input as taught start and goal configurations as well as through "teleoperator" mode, where a user specifies moves directly in the robot's three-dimensional configuration space with a mouse and screen interface. This same scheme has been used to specify three-dimensional Cartesian hand coordinates for the robot's end effector. Experiments with tracking a reflective sphere (position reported by stereo vision) while under reflex control have been conducted. The reflexes have also operated in point-to-point mode in conjunction with on-line planning.

Execution of reflexive collision avoidance has proven valuable in experiments with new servo controllers, with vision-based control and with on-line planning algorithms. In each of these cases, the reflexes have protected the robot and its environment from damage which would otherwise have resulted from coding errors in the experimental higher layers. Even in the case of grossly unstable servo controllers, the reflexes assure the safety of the robot.

Our experimental system and use of reflex control is documented on videotape available from the Center for Automation and Intelligent Systems Research, Case Western Reserve University [14].

8 Summary

In this paper, we consider automatic collision avoidance via artificial reflexes based on acceleration requests. The reflexes operate as a control layer imbedded between a servo layer and actual motor torque commands. By interpreting acceleration requests from the servo controller as equivalent virtual setpoints, the reflexes may evaluate the safety of requests through a search of configuration space. For our experimental system, we have found that the computational efficiency of the reflexes is high enough such that the action of the reflexes is normally transparent. In the event of dangerous input requests, the reflexes refuse to approve the requests, instead substituting a safe response. Use of
the reflexes has aided subsequent research in higher levels of hierarchical control, including time-optimal control, visual servoing and on-line planning.

9 Acknowledgements

This work was supported under grant No. 60NANB9D0934 from the National Institute of Standards and Technology, and by the Cleveland Advanced Manufacturing Program. Their support is gratefully acknowledged.

Fig. 4: Cartesian Tracking Test: Reflexes On

Fig. 5: Cartesian Tracking Test: Reflexes Off
References


[14] Wyatt S. Newman, Recent Advances in Robot Control Research at Case Western Reserve University, Video tape TR-89-139, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, Ohio, June 1989.