Smooth and Flexible Movement Control of a Robot Arm

Chuang Shang-Yi, National Taiwan University, Taiwan
Assigned Laboratory: Motor Intelligence Laboratory
FrontierLab Supervisor: Prof. T. Sugihara, Osaka University, Japan

Abstract— We improved a nonlinear reference shaping controller for manipulators sharing their workspace with humans. The controller is based on the slow and rapid adaptations, which we tried to enhance. After the progress, the slow adaptation can generate movements with smooth endpoint velocity profiles when target position is changed. The rapid adaptation is upgraded as well with respect to not only significantly large external forces but also slight ones. They make the manipulators capable of behaving compliantly to the external forces, and also resuming the motion after the forces are removed, even when the shifts are small. Force detectors are unnecessary in this control system. The validity of the proposed ideas was confirmed via simulations on a planar 4-DOF manipulator.

1 Introduction

Nowadays, the number of robots sharing working space with humans has been increasing, and it is time for us to consider how to make humans feel comfortable when the robots are around. The situation is based on the assumption that it can ease the unpleasantness if the robots are able to act like humans. For example, they should not start moving all of a sudden, which may scare people around it. What we work on is to realize smooth bell-shaped velocity profiles according to biological statistics results[1], and enable the robot arms to be submissive to external forces, which means the flexibility in the topic.

In conventional control researches, robots are usually controlled with explicit inclusions of time parameters, meaning the generated referential motion trajectories are depedent on time. It makes the robots unable to respond to unpredicted events, such as external forces applied from the environment accidentally. Arimoto et al.[2] proposed the virtual spring-damper (VSD) hypothesis that explains human's reaching motion without time-dependent referential trajectories. It is an idea that a parallel pair of mechanical damper and spring draws the endpoint of the robot arm to the target position. A known problem of VSD is that the initial acceleration is maximized, causing an undesired abrupt start of the movement. To fix this, Seto et al.[3] came up with a method named reference shaping. It first generates an intermediate reference position with respect to the current position, the initial position and the target position. Then, by using VSD to connect the endpoint of the robot arm to the reference position, the velocity profile is smoothed into the bell-shape as the initial acceleration is moderated. However, it is based on the idea that the robot arm knows how many percentage it has moved from the the initial point to the target point — if the perturbation makes the arm move out of the straight locus between the start point and the goal point, the arm will fail to estimate the degree of such motion.

To settle this problem, Seto et al.[4] thought out another solution, known as the slow and rapid adaptations. The slow adaptation is for regenerating a smooth trajectory after a new target position is assigned; meanwhile, the rapid adaptation can help the robot resume the movement after external forces are applied. Both of them introduce a concept of changing the initial position adaptively. The slow adaptation lets the initial position keep following the current endpoint position with a few seconds delay, while the rapid adaptation resets the initial position as the current position when external forces make the arm move too far away from the initial position. Although the slow and rapid adaptations can deal with the drawback of reference shaping, it again fails to generate bell-shaped velocity profiles.

This report describes the related work in Section 2, including reference shaping and the slow and rapid adaptations. How the previous problems are solved, and the results of computer simulations are shown in Section 3 and 4 respectively. Finally, the conclusions are summarized in Section 5.

2 Related Work

The schematic block diagram of the proposed controller is illustrated in Fig. 1.

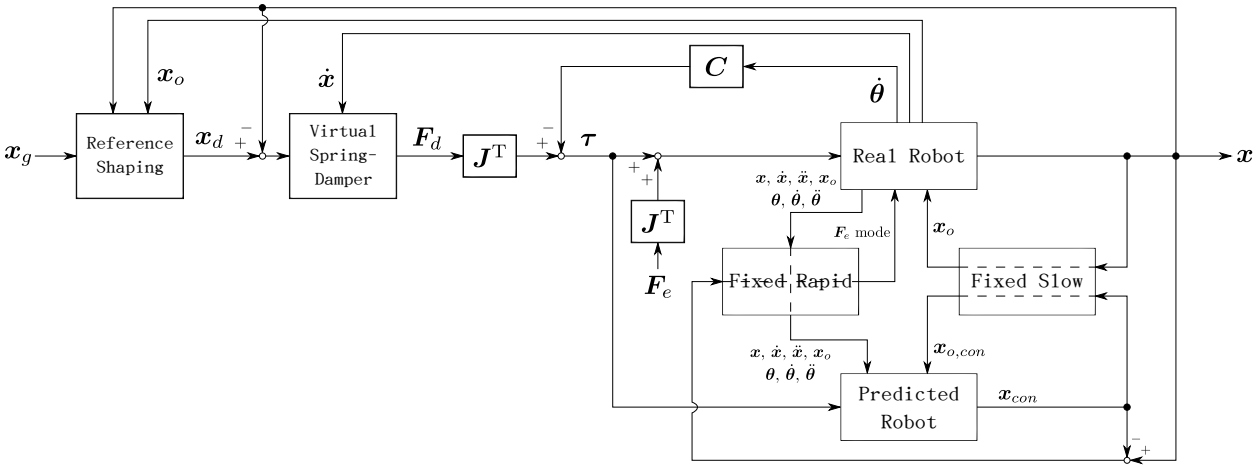

Figure 1: The schematic block diagram of the proposed controller

VSD is applied to a m-joint manipulator in l-dimensional task space. The virtual force $\boldsymbol{F}_d \in \mathbb{R}^l$ and the input joint torque $\boldsymbol{\tau} \in \mathbb{R}^m$ is generated by

$$\boldsymbol{F}_d = \boldsymbol{K}_p(\boldsymbol{x}_d - \boldsymbol{x}) - \boldsymbol{\xi}\dot{\boldsymbol{x}} \tag{1}$$

$$\boldsymbol{\tau} = -\boldsymbol{C}\dot{\boldsymbol{\theta}} + \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{F}_d \tag{2}$$

where $\boldsymbol{K}_p \in \mathbb{R}^{l\times l}$, $\boldsymbol{\xi} \in \mathbb{R}^{l\times l}$, and $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ are the stiffness coefficient matrix, the damping coefficient matrix, and the joint damping coefficient matrix, respectively. $\boldsymbol{x}$, $\dot{\boldsymbol{x}} \in \mathbb{R}^l$ are the position and velocity vectors of the endpoint of the manipulator, and $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}} \in \mathbb{R}^m$ are the joint angle and the angular velocity vectors. $\boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{l\times m}$ denotes the Jacobian matrix with respect to $\boldsymbol{x}$. $\boldsymbol{x}_d$ is the reference input shaped as a function of the current position $\boldsymbol{x}$, the initial position of the endpoint of the manipulator $\boldsymbol{x}_o$, and the target position $\boldsymbol{x}_g$ by using a parameter r as follows:

$$\boldsymbol{x}_d = \{r\boldsymbol{x}_g + (1-r)\boldsymbol{x}\}, \quad 0 < r \leq 1 \tag{3}$$

$$r = 1 - (1-\epsilon)\frac{\|\boldsymbol{x}_g - \boldsymbol{x}\|}{\|\boldsymbol{x}_g - \boldsymbol{x}_o\|}, \quad 0 < \epsilon \ll 1, \tag{4}$$

where ϵ is a constant much less than 1, giving a slight shift from $\boldsymbol{x}_o$ to $\boldsymbol{x}_g$. For the original VSD, r is set to 1.

When the target point is changed, the slow adaptation helps update the initial position close to the current positon, so that the robot can generate a new trajectory from the new initial position to the new destination. An implementation of the slow adaptation is to apply a first-order lag filter to the current position as

$$\mathcal{L}[\boldsymbol{x}_o] = \frac{1}{Ts+1}\mathcal{L}[\boldsymbol{x}], \tag{5}$$

where $\mathcal{L}[.]$ is the Laplace transform denotation, and T is the time constant of the filter meaning how many seconds delay for $\boldsymbol{x}_o$ after $\boldsymbol{x}$.

If external forces move the endpoint far from the supposed working field, the rapid adaptation will handle this situation. To enable the manipulator to restart the movement after the shifts happen, the rapid adaptation operates by

$$\boldsymbol{x}_o = \boldsymbol{x} \quad \text{if} \quad \frac{\|\boldsymbol{x}_g - \boldsymbol{x}\|}{\|\boldsymbol{x}_g - \boldsymbol{x}_o\|} > 1, \tag{6}$$

i.e., once the manipulator is moved further than $\boldsymbol{x}_o$, the rapid adaptation resets $\boldsymbol{x}_o$ for $\boldsymbol{x}$.

3 Methods

When actually applying the slow and rapid adaptations, we found some problems. There are some awkward peaks in the velocity profiles, which mean the movements start abruptly. The reasons are the inconsistency of $\boldsymbol{x}_o$ when $\boldsymbol{x}_g$ changes to another position, and the slight external forces. Therefore, we came up with the following ideas to solve this.

3.1 Fixing Slow Adaptation

When a new target is given, r suddenly changes due to the discontinuity of $\boldsymbol{x}_g$, causing a peak in the acceleration profile.

In order to solve this, we reserve the new assigned target and set $\boldsymbol{x}_g$ as the current $\boldsymbol{x}$ for a tentative target until $\boldsymbol{x}_o$ catches up with $\boldsymbol{x}$. After that, we update $\boldsymbol{x}_g$ by the given position. This can ensure that the velocity converges to zero before the new reaching movement starts, so the peaks in the velocity profiles disappear.

Table 1: The parameter list of 4-DOF manipulator

	Link 1	Link 2	Link 3	Link 4
Link Length l $[m]$	0.10	0.25	0.25	0.15
Position of CoM l_g $[m]$	0.05	0.10	0.10	0.05
Link Mass m $[kg]$	0.5	1.0	1.0	0.5
Moment of Inertia I $[kgm^2]$	0.005	0.01	0.007	0.003

3.2 Fixing Rapid Adaptation

The condition in Eq.(6) of the original rapid adaptation control can only deal with situations that external forces make the endpoint move significantly. If the endpoint is shifted inside the circle centered in $\boldsymbol{x}_g$ with the radius $\|\boldsymbol{x}_g - \boldsymbol{x}_o\|$, the rapid adaptation does not work.

We revise the condition of the rapid adaptation as $\|\boldsymbol{x} - \boldsymbol{x}_{con}\| > \delta$, where *con* means control group in which external forces never exist, and δ is the threshold. The data of the control group are overwritten with the ones of the real robot after external forces are removed, which enables the controller in order to detect the existence of external forces in the future.

4 Results

In this section, results of the simulations with and without the proposed method are both shown. The simulations are based on a planar 4-DOF manipulator illustrated in Fig. 2. The physical parameters of the manipulator model are in Table 1.

$\boldsymbol{\theta}_o = [0.1, 0.3, 0.6, 1.0]^{\mathrm{T}}$ [rad] is used as the initial joint angle vector, while $\boldsymbol{\theta}$ is the accumulated angle vector.

The sampling time is 1 [ms], and T of the delay filter in Eq.(5) is set to 0.33 [s]. We choose $\boldsymbol{K}_p = \mathrm{diag}\{500, 500\}$, $\boldsymbol{C} = \mathrm{diag}\{0.1, 0.1, 0.1, 0.1\}$, and $\boldsymbol{\xi} = \mathrm{diag}\{50, 50\}$ for Eq.(1) and Eq.(2). In Eq.(4), $\epsilon = 0.001$ is used.

4.1 Results of Fixing Slow Adaptation

For checking the slow adaptation, the goal positions are set for $\boldsymbol{x}_{g1} = [0.5, 0.3]^{\mathrm{T}}$, $\boldsymbol{x}_{g2} = [0.3, 0.3]^{\mathrm{T}}$, $\boldsymbol{x}_{g3} = [0.5, 0.1]^{\mathrm{T}}$ and $\boldsymbol{x}_{g4} = [0.3, 0.1]^{\mathrm{T}}$ [m].

Fig. 3 and Fig. 4 are the velocity profiles before and after fixing the slow adaptation, respectively. We can easily tell that it takes longer time to achieve all the goals in the latter profile. The reason is that we wait for $\boldsymbol{x}_o$ catching up with $\boldsymbol{x}$.

The original slow adaptation changes $\boldsymbol{x}_g$ to $\boldsymbol{x}_{g_{n+1}}$ immediately after $\boldsymbol{x}_{g_n}$ is reached, causing a sudden start when the manipulator begins to move toward

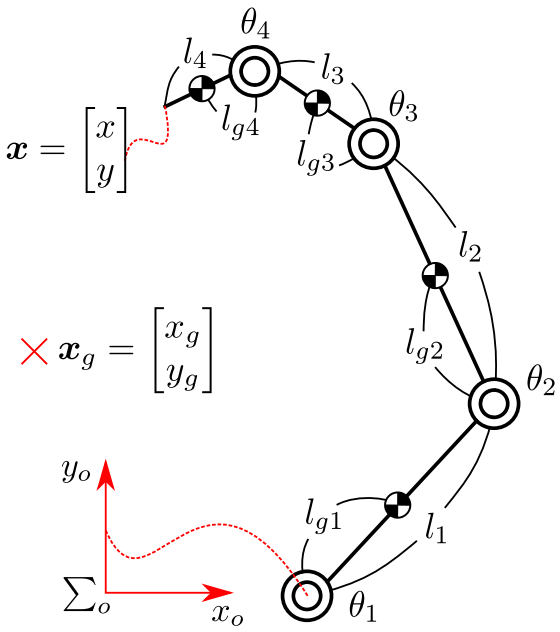

Figure 2: The simulation setup

$\boldsymbol{x}_{g_2}$ at 1.8 [s] in Fig. 3. On the other hand, Fig. 4 demonstrates that the manipulator successfully realized smooth movements in the bell-shapes.

4.2 Results of Fixing Rapid Adaptation

When checking the rapid adaptation, the external force $[-2, 2]^{\mathrm{T}}$ [N] is applied from 1 [s] to 1.2 [s]. The tolerance δ set to fix the rapid adaptation is 1 [mm].

According to Fig. 5, with the original slow adaptation, the manipulator fails to detect the external force which makes a peak at 1.2 [s], and keeps going to the target point. After fixing the rapid adaptation, in Fig. 6, we can see that once the perturbation exists, the controller tries to create a virtual force, intending to balance the external force. When the external force is removed, the manipulator first generates a zero velocity, and then, resumes a smooth movement toward the target position.

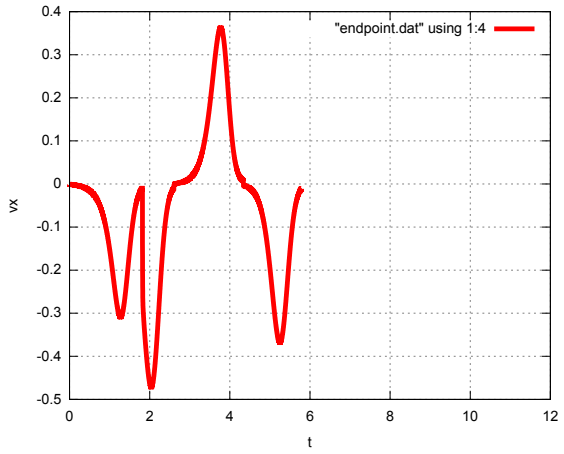

Figure 3: The velocity profile before fixing the slow adaptation (Sequential targets)

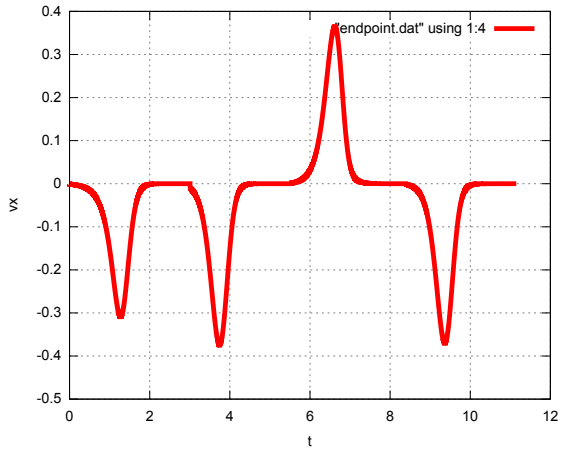

Figure 4: The velocity profile after fixing the slow adaptation (Sequential targets)

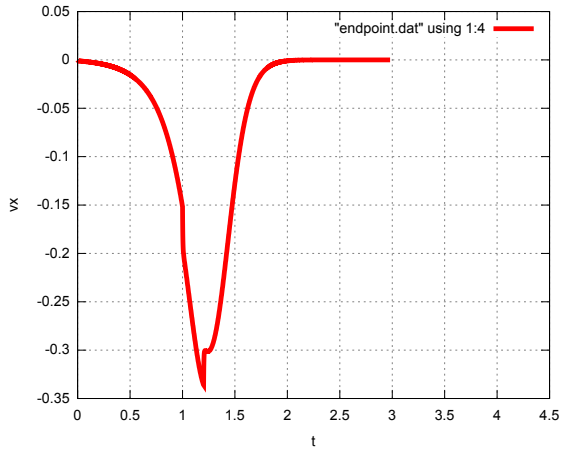

Figure 5: The velocity profile before fixing the rapid adaptation (The external force occurs from 1 [s] to 1.2 [s])

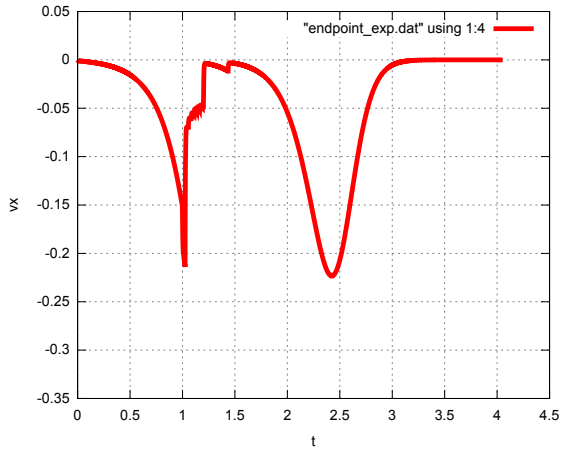

Figure 6: The velocity profile after fixing the rapid adaptation (The external force occurs from 1 [s] to 1.2 [s])

4.3 Discussion

With the proposed ideas, the result in Fig. 7 shows that discontinuity of $\boldsymbol{x}_o$ only happens when external forces occur.

Due to the existence of ϵ, in Fig. 8, there is always a peak of acceleration for initial actuation at the moment the new reaching movement starts, while the velocity profiles are almost beautiful bell-shaped as Fig. 4 describes. We think we can neglect the peaks because there seems no abrupt restart during the movements.

These simulation results verify the effectiveness of the proposed ideas. After the improvements, the slow adaptation becomes able to generate smooth endpoint velocity profiles when handling sequential targets, while the rapid adaptation is capable of resuming the movement interrupted by external forces applied from the environment even when the forces are weak.

5 Conclusion

The slow and rapid adaptations described in this report are based on the nonlinear reference shaping controller and the virtual spring-damper hypothesis. With these previous studies, explicit time inclusions and force detectors are not required for the proposed method. The manipulator can generate smooth and flexible movements with only the data of $\boldsymbol{x}$, $\boldsymbol{x}_g$ and $\boldsymbol{x}_o$. The slow adaptation is used for updating the initial position of the manipulator, and the rapid adaptation is only motivated when external forces occur.

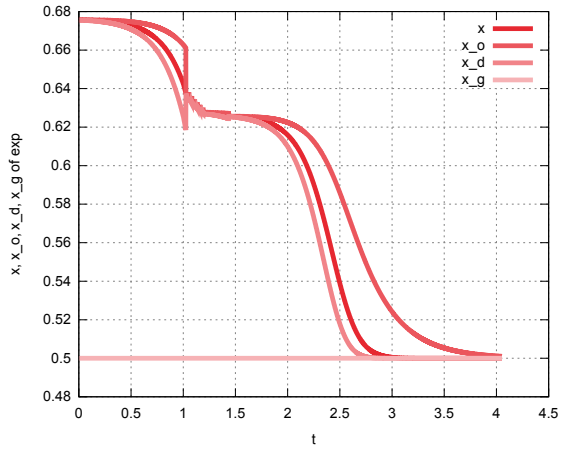

Figure 7: The position data after fixing the rapid adaptation (The external force occurs from 1 [s] to 1.2 [s])

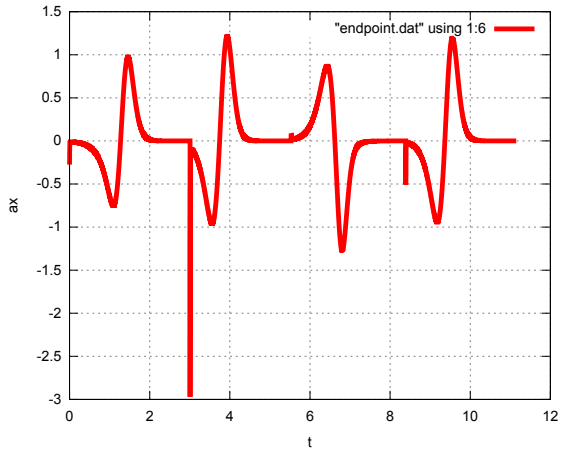

Figure 8: The acceleration profile after fixing the slow adaptation (Sequential targets)

In this research, we eliminated the drawbacks of the slow and rapid adaptations. The improved slow adaptation helps the robot deal with the restarts when reaching to sequential targets, as the rapid adaptation is enhanced to be able to sense the slight shifts which could not be detected before.

Acknowledgment

I would like to show my gratitude to Prof. Tomomichi Sugihara and the members of Motion Intelligence Laboratory for their extraordinary supports. These achievements would have not been possible without all the encouragements and the insightful comments. Thanks for the wonderful collaboration.

The assistance offered by JASSO scholarship is also remarkable. I strongly benefited from and deeply appreciate it.

Last but not least, I am immensely grateful to that National Taiwan University and Osaka University giving me this opportunity to study in Japan with the FrontierLab program.

References

[1] W. Abend, E. Bizzi, and P. Morasso. Human Arm Trajectory Formation. In *Brain*, Vol. 105, pp. 331–348, 1982.

[2] S. Arimoto and M. Sekimoto. Human-like Movements of Robotic Arms with Redundant DOFs: Virtual Spring-Damper Hypothesis to Tackle the Bernstein Problem. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, pp. 1860–1866, 2006.

[3] F. Seto and T. Sugihara. Online Nonlinear Reference Shaping with End-point Position Feedback for Human-Like Smooth Reaching Motion. In *9th IEEE-RAS International Conference on Humanoid Robots*, pp. 297–302, 2009.

[4] F. Seto and T. Sugihara. Motion control with slow and rapid adaptation for smooth reaching movement under external force disturbance. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1650–1655, 2010.