A Real-time Motion Data Reduction and Restoration Compatible with Robot's Physical Limits

Doik Kim, Jung-Min Park and Seokmin Hong
Interaction and Robotics Research Center, KIST,
Hwarangno 14-gil 5, Seongbuk-gu, Seoul, 136-791, Republic of Korea

Keywords: Data Reduction, Restoration, Hardware Limit, Convolution Interpolation.

Abstract: In order to control a robot, motion data obtained from various interface devices such as haptic and tele-operating interfaces, and a motion capture system, or from predefined parametric equations are needed. It is hard to store and transmit the data because of their high control frequency and synchronization problem. In addition, usually, all these data are only for one target robot system, i.e., it is hard to use them to other robot system with different hardware limits. In order to solve the problems, the reduction of the original data is conducted with two stage magnitude quantization and the restoration without violating the hardware limits, such as maximum velocity, acceleration, etc., is conducted with the convolution interpolation, in this paper. The proposed method can be used in on-line data transmission and for off-line data storage.

1 INTRODUCTION

A robot is controlled by motion data obtained from various interface devices of tele-operation, haptics, and motion capture system as well as parametric equations for trajectory generation. If motion data are to be stored for replaying later or to be transmitted to remote devices in real-time, several problems need to be considered. Especially the realtimeness, and target system's physical limits are essential to be considered for stable and safe control.

Data size for controlling a robot is different from robot to robot, according to its control Hz, DOF(Degree of Freedom), control variables, etc. Absolute size of motion data is less than other types of data such as audio and video, but it must be guaranteed a real-time synchronization from sensors to actuators through networks within its control sampling time. This hard real-time condition makes load of the network comparable to audio and video data(Hinterseer et al., 2008).

In addition, the transmitted signal should not violate physical hardware limits such as velocity, acceleration, jerk, etc. In many cases, the transmitted data will be used for the same robot hardware system, but if motion data are stored as reference data, they can be applied to other robot hardware systems as far as they are compatible to the original robot. Compatibility can be qualified by considering workspace, DOF, structural similarity, and actuation hardware limits, etc. For example, if two robots have similar workspace and motion data for control are composed of the position and orientation of the end-effector, there is high possibility that the motion data are compatible between two robots regardless of their structures and DOF from the theoretical viewpoint. However, in order to share the motion data between real robot systems, the restored data should not violate physical hardware limits of a target robot; otherwise they make the robot unstable. Thus, non-violation of physical hardware limits is a fundamental condition for safe and stable control.

In order to solve the realtimeness, data reduction/restoration methods have been studied. Most research is focused on the haptic data reduction and transmission. DPCM(differential pulse code modulation) related methods are used to reduce haptic motion data by (Shahabi et al., 2002; Kron et al., 2004). Hirche, et al., proposed a psychophysically motivated compression method with the deadband concept to reduce the haptic data(Hirche et al., 2007; Hirche and Buss, 2007). If the difference between the current data and the most recently transmitted data exceeds a predefined perception threshold (the deadband), the current data are transmitted, and otherwise the haptic data are abandoned. It is extended to 3 DOF haptic devices in (Hinterseer et al., 2008), and 6-DOF in (Sakr et al., 2010; Sakr et al., 2011) A haptic data reduction

Kim D., Park J. and Hong S..
A Real-time Motion Data Reduction and Restoration Compatible with Robot's Physical Limits.
DOI: 10.5220/0005041303600367
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 360-367
ISBN: 978-989-758-040-6

based on human perception is summarized in (Steinbach et al., 2011). All these are concerned on the realtimeness by reducing motion data, especially in telepresence and tele-action system, and thus human perception is an important aspect in reducing data. However, from the viewpoint of control, human perception is a sub-condition of motion data reduction, i.e., motion data can be reduced more beyond the human perception to control a robot, and it has not been considered deeply.

In the case of restoration, physical hardware limits are not considered importantly, because the data is used for the same hardware. However, if a set of motion data is transmitted to or stored for later usage of several different robots, physical hardware limits must be considered for safe and stable operation. This discordance between source motion data and target robots cannot be solved in all cases, but in many cases, it can be relaxed and increase the compatibility in restoration stage by considering hardware limits.

To solve these realtimeness and hardware compatibility problems, two-stage data reduction and convolution-based data restoration are proposed as follows: section 2 and 3 describe the reduction and restoration method, respectively. Section 4 shows experimental results, and section 5 concludes this paper.

2 DATA REDUCTION WITH MINIMAL SHAPE INFORMATION

Motion data are considered as set of signals having similar properties such as noise level, sampling time, etc. Specifically, motion data such as position, orientation, joint angles, which are mostly used for controlling a robot and feeding the status back to a user, are considered in the paper and other kinds of data can also be considered similarly without any problem.

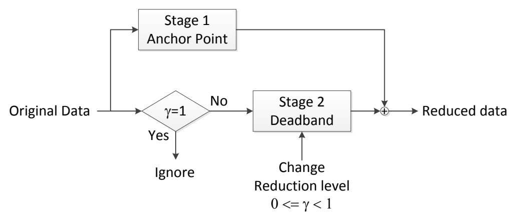


Figure 1: Flow chart for data reduction: The original data is reduced by two-stage process.

The reduction is conducted in two stages: the anchor point and the deadband as shown in Figure 1. The first stage finding the anchor point is to preserve the minimal shape of motion data and the second stage is to reduce motion data by adjusting the deadband size.

Before reducing motion data, a critical length is introduced to normalize the data for removing dependency of kinematic differences. The max length, $d_{\max}$ is defined as a constant critical length of a robot system. It can be any value representing a maximum magnitude of a signal. For example, a fully extended link length of a robot can be one candidate for position signal, and maximum joint angle can be a candidate for joint angle signal. The max length $d_{\max}$ will be used as scale factor in reduction and restoration, and thus it should not be changed in all process.

2.1 Anchor Point

With the points of zero velocity which will be called the anchor point, we can obtain several information of a signal such as direction change, start and stop point. Set of anchor points are minimum set of motion data to represent the overall shape of motion.

The anchor point can be obtained by finding a point with zero velocity, but it is difficult to find the point, because the original data are discretized. So, the zero crossing condition is used to find an approximate point of zero velocity as follows:

$$y_a = x(k), \text{ if } v(k) \cdot v(k-1) \leq 0 \tag{1}$$

where y_a is an approximate anchor point. $x(k)$ is a point and $v(k)$ is the velocity of the point $x(k)$ of the original data, respectively. If Eq. (1) satisfies, a point somewhere between $x(k-1)$ and $x(k)$ has zero velocity, and thus the current point $x(k)$ is chosen for an approximate anchor point.

In addition, the velocity, $v(k)$ in Eq. (1) is obtained usually from the numerical differentiation of points $x(k)$, and this makes noise in velocity. To remove effects of the noise, the following condition is added

$$y_{i+1} = \begin{cases} y_a, \text{ if } \|y_a - y_i\| \geq d_{\min} \\ \text{none, otherwise} \end{cases} \tag{2}$$

where $d_{\min}$ is the minimum anchor distance between the previous reduced point y_i and the new reduced point y_{i+1}. Note that here the reduction point y_i is composed of only the anchor points, y_a. The range of the minimum anchor distance is $0 \leq d_{\min} \leq d_{\max}$.

From Eq. (2), we can set a minimum distance between adjacent anchor points. If $d_{\min} = 0$, all the anchor points, y_a, in Eq. (2) will be the reduced point y_{i+1}, and if $d_{\min} = d_{\max}$, almost all the anchor points will be abandoned and there will be the least reduced point and loose the shape of data in most cases.

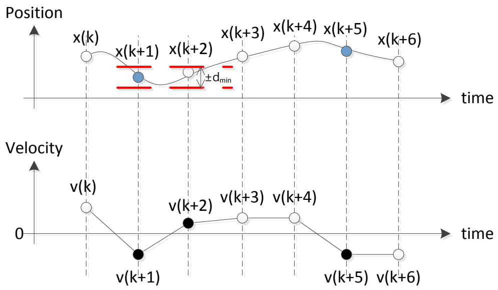


Figure 2: The anchor point is defined a point with zero velocity but for discretized points, an approximate anchor point is obtained.

Figure 2 shows the relation of anchor points. Somewhere between $v(k)$ and $v(k+1)$ has zero velocity and thus $x(k+1)$ is set as the anchor point, represented as a black circle in velocity graph and is assumed to be new reduced point represented as a blue circle in position graph. The next $v(k+2)$ also crossed zero, and thus $x(k+2)$ is a candidate of the anchor point but it is within the boundary of $d_{\min}$, so it is abandoned.

2.2 Point with Deadband

The basic idea of the deadband has been studied by many researchers as in (Hirche et al., 2007; Hirche and Buss, 2007). However, in this paper, previous method is extended to adjust the reduction level from no reduction to maximum reduction.

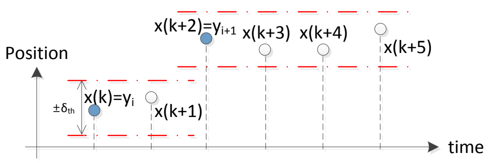


Figure 3: Reduction by deadband: If $x(k+i)$ is within the threshold, i.e., $|x(k+i) - y_i| \leq \delta_{\text{th}}$, it will be abandoned, otherwise, it will set as new transmission point y_{i+1}.

The reduction law is simple by quantization of data with length comparison as follows:

$$\mathbf{y}_{i+1} = \begin{cases} \mathbf{x}(k), \text{ if } \|\mathbf{x}(k) - \mathbf{y}_i\| \geq \delta_{\text{th}} \\ \text{none, otherwise} \end{cases} \quad (3)$$

where $\mathbf{x}(k)$ is the kth original data vector, $\mathbf{y}_i$ is the ith reduced data. Figure 3 explains the deadband approach. A blue circle represents reduced data, $\mathbf{y}_i$, and white circle represents abandoned original data.

The threshold value is defined as

$$\delta_{\text{th}} = \gamma d_{\max} \quad (4)$$

The reduction level, γ is a value in the range of $0 \leq \gamma \leq 1$ representing fraction of $d_{\max}$.

The threshold, δ_{th} is automatically calculated by the user adjusted reduction level γ and the max length, $d_{\max}$. As shown in Eq. (3), a new reduced data $\mathbf{y}_{i+1}$ is defined if an error between the current original data $\mathbf{x}(k)$ and the current reduced data $\mathbf{y}_i$ is larger than the given threshold, δ_{th}. Otherwise, no data will be stored or transmitted.

Note that, the threshold represents the maximum quantization error between the original and reduced data. The reduced data $\mathbf{y}_{i+1}$ has maximum error of δ_{th}, i.e., $\|\mathbf{y}_{i+1} - \mathbf{x}(k)\| \leq \delta_{\text{th}}$ where $\mathbf{x}(k)$ is set of original data between $\mathbf{y}_i$ and $\mathbf{y}_{i+1}$.

By adjusting γ, a user can change relative reduction ratio which is related to the error range. If $\gamma = 0$, i.e., $\delta_{\text{th}} = 0$, there is no error between the reduced data and the original data, i.e., reduction is not occurred. If $\gamma = 1$, i.e., $\delta_{\text{th}} = d_{\max}$, almost all data are vanished because $d_{\max}$ is a maximum magnitude of a signal.

In the case of maximum reduction, $\gamma = 1$, it is, however, inconvenient if no data are stored or transmitted, because we don't have any information of the original data. To overcome this, the maximum reduction is slightly modified to represent set of data with minimal information of the signal by using the anchor point described in the previous section.

In summary, the proposed algorithm has three user defined parameters, the maximum critical length, $d_{\max}$, the reduction level, γ, and the minimum anchor distance, $d_{\min}$. The maximum critical length, $d_{\max}$ is a constant and is related to the mechanical specification of a robot system. This value is also used in restoration stage. The reduction level γ is a user defined relative reduction ratio which related to the quantization error between the original data and the reduced data. The minimum anchor distance, $d_{\min}$ is a user defined value, and is mostly related to the signal properties such as signal noise and numerical noise. The reduction level, and the minimum anchor distance can be adjusted any time, even in running the algorithm. Note that signals for the reduction are a position and its derivative (velocity) and thus these signals should be obtained appropriately before reduction.

3 DATA RESTORATION WITHOUT VIOLATING PHYSICAL LIMITS

In the previous section, motion data are reduced with magnitude quantization. The restoration process is to generate a trajectory satisfying the hardware system of a target robot from randomly transmitted data in real time.

As shown in the previous works(Hinterseer et al., 2008), they used a zero-order-hold strategy or the linear prediction method to restore reduction data. A zero-order-hold strategy holds a received sample until a new sample arrives, and the linear prediction method uses an average change-of-rate from the previously received samples to extrapolate current sample. All these methods and other prediction methods always give discontinuity when a new sample arrives, since the prediction cannot predict future motion exactly. From the aspect of control, discontinuity makes system unstable.

In addition to this discontinuity one more severe problem not mentioned in the previous research is that the hardware limit. The reduced data can be applied to any robot with different hardware specifications. For example, two robots have the same mechanical structure but their actuator limits such as acceleration, velocity and position limit can be different from each other. In this case, the reduced data should be restored to signals that are not violating the hardware limits of each robot.

In order to solve these problems, a convolution interpolation is proposed to obtain continuous and hardware-compatible motion data in real time.

3.1 Convolution Interpolation to Generate a Trajectory

The convolution interpolation has several useful aspects: it can generate continuously differentiable trajectories without violating physical system limits by applying successive convolution operation; a recursive form of discrete convolution can reduce computational loads drastically. In this subsection, the convolution interpolation to generate a trajectory shown in (Lee et al., 2011; Lee et al., 2012) is summarized briefly.

Let us suppose that $y_0(t)$ is an non-convoluted input function defined in $[0,t_0]$, $h(t)$ is a rectangular function of unit area defined in time duration $[0,t_h]$, and $y_1(t)$ is the 1st output function convoluted by two functions $y_0(t)$ and $h(t)$. Then the convolution has following properties:

1. The final time t_1 of $y_1(t)$ is the sum of both time durations of $y_0(t)$ and $h(t)$, i.e., $t_1 = t_0 + t_h$.
2. The area of output $y_1(t)$ is always equal to that of input, $y_0(t)$.
3. The maximum absolute value of $y_1(t)$ is always less than or equal to that of $y_0(t)$. Especially, if the input maintains a constant over the time duration t_h, then output reaches the input value.

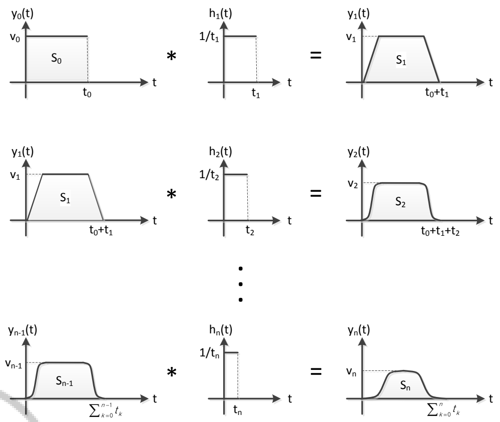


Figure 4: The effect of the convoluted trajectory with non-zero final condition is shown. The trajectory becomes smoother with more convolutions.

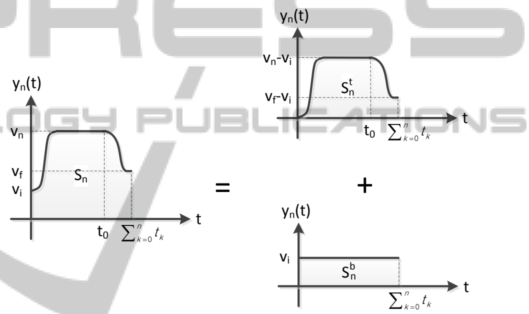


Figure 5: A general trajectory with non-zero initial and final velocity can be calculated by decomposing the trajectory into non-zero initial and non-zero final condition.

With the above properties, convolution operation can be used for generating a trajectory as shown in Figure 4. The initial and the final velocity are assumed zero, $v_i = v_f = 0$, and v_k is the kth convoluted velocity. S_k are the area calculated with the kth convolution and are equal regardless of k from the property 2. After the 1st convolution, the input $y_0(t)$ function which is a class C^{-1} function becomes a continuous but not a differentiable function $y_1(t)$, i.e., $y_1(t)$ becomes a C^0 function. After the 2nd convolution, the input $y_1(t)$ function becomes a function $y_2(t)$ which has a continuous first-derivatives, i.e., a C^1 function, and so on. Consequently, by increasing the number of convolution, we can increase the order of differentiability as many as we want, and it can be represented as $y_k(t)$ is a C^{k-1} function. In convolution, the completion time is also increased from the property 1.

Figure 5 shows the case of non-zero initial and final velocity. The area S_n is divided into two area: a translated area S_n^t with zero initial velocity and a rectangular base area S_n^b with the initial velocity, v_i. By adding two area, the entire area can be obtained,

$S_n = S_n^t + S_n^b$.

In order to guarantee that the resultant function does not violate system limits, the time duration t_k of a unit-area function $h_k(t)$ becomes

$$t_k = \frac{v_{\max}^{k-1}}{v_{\max}^k} \text{ for } k = 1, 2, \cdots, n \tag{5}$$

where $v_{\max}^k$ denotes the system limits of kth convolution. For example, $v_{\max}^0$, $v_{\max}^1$, and $v_{\max}^2$ denote the maximum velocity, acceleration, jerk, respectively. By using Eq. (5) and from property 3, we can guarantee the convolution interpolation does not violate the system limits.

From Figure 5, total moving distance is represented as follows:

$$S_n = S_n^t + S_n^b = v_0\, t_0 + \frac{(v_f + v_i)}{2} \sum_{k=1}^{n} t_k \tag{6}$$

There are two cases in using Eq. (6) with known S_n : 1) if $v_0 = v_{\max}^0$, then the only unknown is t_0, 2) if the final time t_f is known, then t_0 can be obtained from property 1 and Eq. (5), and thus the only unknown is v_0. These relations can be solved as follows:

Max Velocity. If $v_0 = v_{\max}^0$, then we can arrive S_n as fast as possible and time t_0 for this can be obtained as follows from Eq. (6)

$$t_0 = \left(S_n - \frac{(v_f + v_i)}{2} \sum_{k=1}^{n} t_k \right) / v_{\max}^0 \tag{7}$$

Known Final Time. If we know the final time, t_f to go, then t_0 can be obtained from Property 1 as follows:

$$t_0 = t_f - \sum_{k=1}^{n} t_k \tag{8}$$

Note that t_0 is equal to zero or negative if given time interval, t_f is too short. This means that it is impossible to reach the target with the system limits within the given time interval. For positive t_0, v_0 is obtained as follows:

$$v_0 = \left(S_n - \frac{(v_f + v_i)}{2} \sum_{k=1}^{n} t_k \right) / t_0 \tag{9}$$

So far convolution operation is described in the continuous time domain, but for fast and simple calculation, it can be expressed in the discrete time domain as follows: The convolution sum considered with nth unit-area function, $h_n[k]$ for $0 \le k \le m_n - 1$ can be expressed by a recursive form as follows:

$$y_n[k] = \frac{y_{n-1}[k] - y_{n-1}[k - m_n]}{m_n} + y_n[k-1] \tag{10}$$

where k and m_n are positive integers satisfying $k = [t/T_s]$ and $m_n = [t_n/T_s]$, respectively, with sampling time T_s and Gauss floor function $[x]$ to denote the largest integer not greater than x.

3.2 Continuously Restored Trajectory with Convolution

In the previous subsection, a convolution interpolation is introduced and will be used to restore the reduced motion data. Without loss of generality, we only use the 1st convolution which preserves up to the acceleration. If more hardware limits are supposed to be satisfied, more order of convolution is needed, and the relations can be obtained similarly.

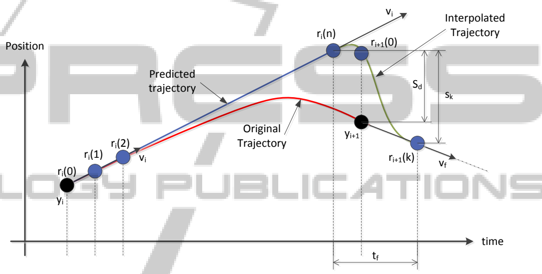


Figure 6: When a new point is arrived, there is a gap, S_d, between the predicted point, $r_i(n)$ and the arrived point, y_{i+1}. This gap is interpolated by using the convolution method as fast as possible without violating system limits.

In restoring data, a linear prediction is used, i.e., when a reduced point is composed of the point $\mathbf{x}(i)$ and the velocity $\mathbf{v}(i)$. With these information, the restored point, $\mathbf{r}_i(k)$ becomes

$$\mathbf{r}_i(k) = \mathbf{v}(i) \cdot (k\Delta t) + \mathbf{x}(i) \tag{11}$$

where i and k are indexes for the ith transmitted data, and kth restored data, respectively. Δt is the sampling time which may be different from the reduction side. In the case that the velocity is not available, the last transmitted point can used for restored data, i.e., $\mathbf{r}_i(k) = \mathbf{x}(i)$, and this will give more errors than the linear prediction case.

As shown in Figure 6, the gap between the current predicted point to the newly transmitted point is interpolated by the convolution method as fast as possible. In the figure, the linear prediction is used with the transmitted instantaneous velocity. It is also possible to use a zero-order-holder strategy or any other prediction methods, but the important thing is the gap between the current point and the newly transmitted point.

S_d in Figure 6 is the distance between the current point and the newly transmitted point, but it is hard to reach on the newly arrived point within one sampling time in general, and thus we need more time. For the linear prediction case as shown in Figure 6, we need to go more distance S_k calculated as follows:

$$S_k = S_d + v_f t_f \tag{12}$$

where v_f is the transmitted final velocity, and t_f is the time for reaching out to S_k. The distance S_k can also be calculated as in Eq. (6) and by equating with Eq. (12),

$$S_d + v_f t_f = v^0_{\max}\, t_0 + \frac{(v_f + v_i)}{2} t_1 \tag{13}$$

where v_0 in Eq. (6) is replaced with the maximum velocity, $v^0_{\max}$, and $t_f = t_0 + t_1$ from property 1. v_i is the velocity of the current point. Finally, by solving the above equation, t_0 becomes

$$t_0 = (s_d - \frac{v_i - v_f}{2} t_1)/(v^0_{\max} - v_f) \tag{14}$$

where $t_1 = v^0_{\max}/v^1_{\max}$ as in Eq. (5). Note that if the linear prediction is not used, then $S_k = S_d$ and thus the term for the linear prediction $v_f t_f$ becomes zero in Eq. (13).

Now, we know all variables to interpolate between the current point, $r_i(n)$, and the newly transmitted point, y_{i+1}. By applying the convolution interpolation, we can connect the points smoothly and quickly without violating the hardware limits. Note that the error between the current point and the newly transmitted point is reduced as fast as possible with the maximum velocity and acceleration. Note also that the signal distortion almost always exists whenever a new point is transmitted, and it is related to the threshold value, δ_{th}.

4 EXPERIMENTS

In order to verify the proposed reduction/restoration method, real motion data are obtained from a 6 DOF arm attached on the humanoid Mahru developed by KIST. The maximum length $d_{\max}$ is set to $1m$ for convenience although the actual length is $0.72m$ when the arm is unfolded straightly.

Arbitrary motions are conducted to analyze the performance. In this paper, we only considered the position (x,y,z) of the end-effector. Other values such as joint angles can be considered similarly.

4.1 Data Reduction

The position signal has three elements (x,y,z), but for calculating the anchor points, each signal is calculated independently, because the anchor point represents the stationary point with zero velocity. If all three elements are considered together as a vector, each signal loses its shape information. Figure 7 represents the anchor points of the y element. The anchor points, green points in the figure, indicate points where the direction is changed and they are the minimum number of points to present the shape of original signal as shown in the figure with the blue line. In calculating the anchor point, The minimum anchor distance is set as $d_{\min} = 0.005m$. The original data has 2078 points and the number of anchor points is 10 for y element.

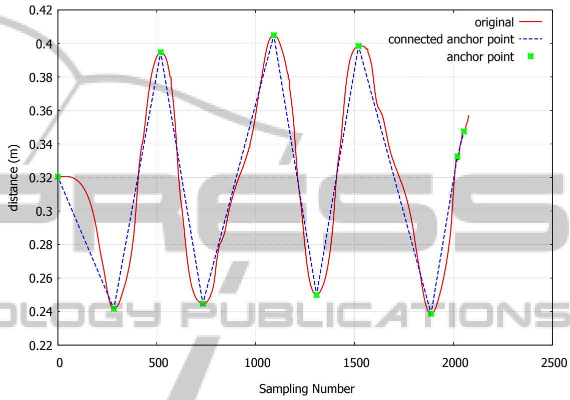


Figure 7: Anchor points of y element. The red line represents the original signal and discontinuous points on the blue line represent the anchor points.

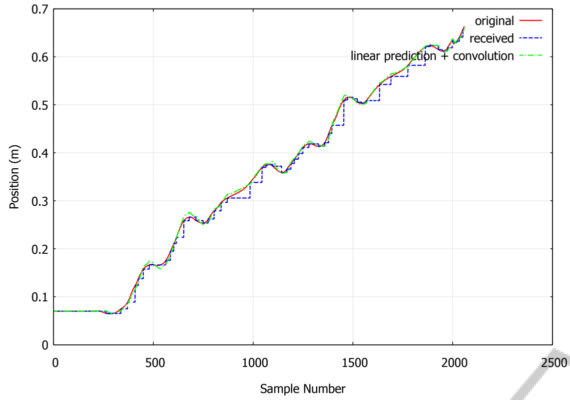
For the deadband approach, the distance is calculated with the position vector unlike the anchor point, because relative relations of each element are important. The blue line in Figure 8 represents the reduction results. The reduction ratio for this case is 4.23% (88 / 2078) with the linear prediction.

The error due to the reduction is shown in Figure 9(a). The error can be estimated by the reduction level γ. In this case, $\gamma = 0.01$ is used and thus the maximum error is $\gamma d_{\max} = 0.01m$, which is valid as shown in Figure 9(a).

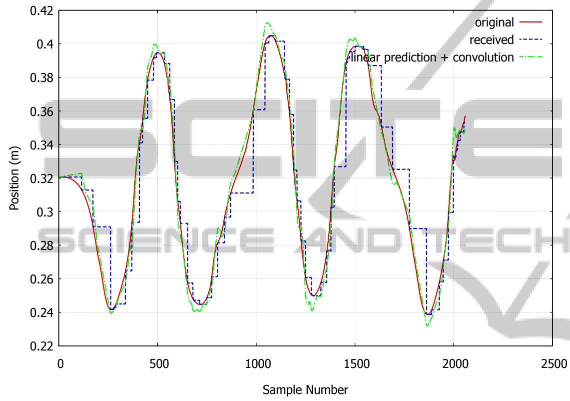
4.2 Data Restoration

The green line of Figure 8 shows the restoration results with the convolution interpolation and the linear prediction. As shown in the figure, all the discontinuity points are connected smoothly. If more smoothness is needed, we only need more convolutions which need more delay in restoration.

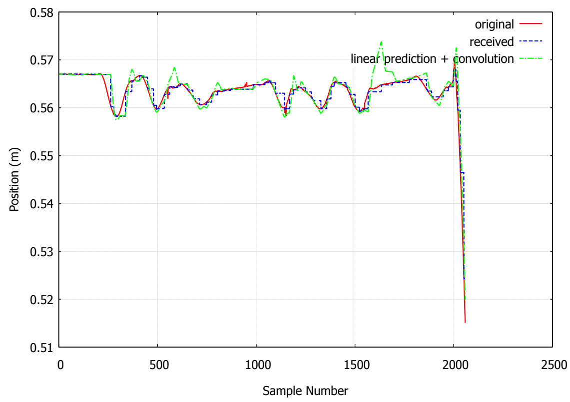
The error level of the restored data shown in Figure 9(b) is similar to the reduction error shown in Figure 9(a). In most cases, the restoration error level is equal to the reduction error, because the role of the convolution interpolation is to connect the discontinuity, i.e., it does not give additional error. However, in



(a) x



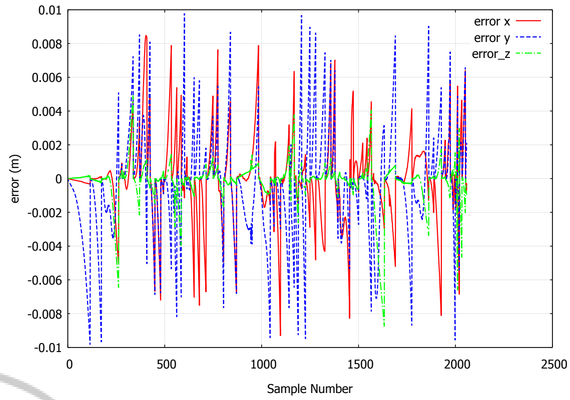
(b) y



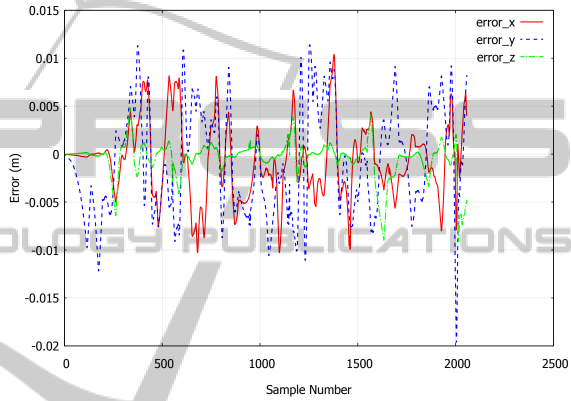
(c) z

Figure 8: Results of data reduction and restoration. The red line is for the original signal, the blue line for the reduced signal and the green line for the restored signal.

some cases, there are larger errors. This comes from the combination of wrong prediction, numerical error, convolution interpolation, and different hardware limits. For example, the convolution interpolation gives a different trajectory from the original one in direction change to connect points smoothly. But in most cases, the convolution interpolation gives desired results if the hardware limits are not so poor.



(a) reduction



(b) restoration

Figure 9: Error of data reduction(a) and restoration(b). All errors in reduction and restoration are related the reduction level γ and the maximum critical length $d_{\max}$.

The green line in Figure 10 shows the result of the convolution when the maximum velocity is 0.1 times less than that of the original system. In this case, as shown in the figure with red circles, the system cannot follow the original signal because it has lower maximum velocity, but the restored trajectory is generated with the maximum velocity to follow the given original trajectory as fast as possible. The convolution interpolation gives a smooth trajectory even in the case of a low performance system, and this property is very important for the safety aspect.

Note that the proposed method is to reduce and restore a signal, and thus it does not have any limit of signal type, but if a signal is nosy such as force and acceleration, we need to choose parameters carefully. In addition, derivate of the signal is also needed in reduction stage, and this magnifies the noise too. Consequently, in order to improve the performance, the relation between the parameters and signal noise needs to be analyzed further.

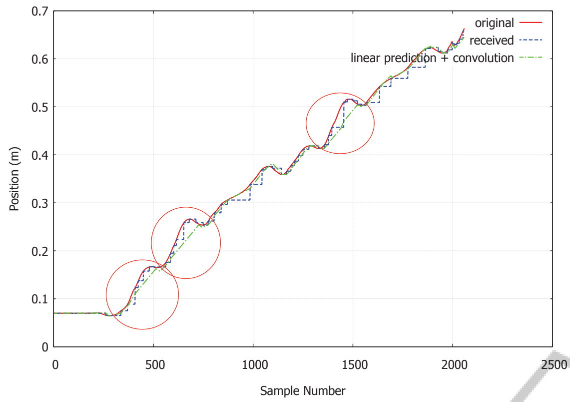


Figure 10: Restoration results when the maximum velocity is less than the original one. At several places enclosed by red circles, the green line cannot follow the red line because of the hardware limits.

5 CONCLUSIONS

In reduction and restoration of motion data, most researchers focused on the reduction and not on the restoration. However, the restoration is also important because of the difference of hardware limits. In the paper, we suggested a convolution based motion data restoration method to restore data without violating the hardware limits and to generate a smooth trajectory in real-time. In addition, we can expect error level in reduction and restoration by using the proposed method. With only 4.23% of the original data, we can restore the signal with the error level of $0.01m$. The proposed method can be used in any tele-operation and tele-presence system and for the data storage. Especially, the proposed method is useful for a poor communication environment, heterogeneous master-slave system, and simultaneous control of multi-robot.

ACKNOWLEDGEMENTS

This work is supported by the KIST institutional program (Project No. 2E24800).

REFERENCES

Hinterseer, P., Hirche, S., Chaudhuri, S., Steinbach, E., and Buss, M. (2008). Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 56(2):588–597.

Hirche, S. and Buss, M. (2007). Transparent data reduction in networked telepresence and teleaction systems. part ii: Time delayed communication. *Presence: Teleoperators Virtual Environment*, 16(5):531–542.

Hirche, S., Hinterseer, P., Steinbach, E., and Buss, M. (2007). Transparent data reduction in networked telepresence and teleaction systems. part i: Communication without time delay. *Presence: Teleoperators Virtual Environment*, 16(5):523–531.

Kron, A., Schmidt, G., Petzold, B., Zah, M. F., Hinterseer, P., and Steinbach, E. (2004). Disposal of explosive ordnances by use of a bimanual haptic telepresence system. In *IEEE International Conference on Robotics and Automation*, pages 1968–1973.

Lee, G., Kim, D., and Choi, Y. (2012). Faster and smoother trajectory generation considering physical system limits under discontinuously assigned target angles. In *IEEE International Conference on Mechatronics and Automation*, pages 1196–1201, Chengdu, China.

Lee, G., Yi, B.-J., Kim, D., and Choi, Y. (2011). New robotic motion generation using digital convolution with physical system limitation. In *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 698–703, Orlando, FL, USA.

Sakr, N., Georganas, N. D., and Zhao, J. (2010). Exploring human perception-based data reduction for haptic communication in 6-dof telepresence systems. In *IEEE International Symposium on Haptic Audio-Visual Environments and Games (HAVE)*, pages 1–6, Phoenix, AZ.

Sakr, N., Georganas, N. D., and Zhao, J. (2011). Human perception-based data reduction for haptic communication in six-dof telepresence systems. *IEEE Trans. on Instrumentation and Measurement*, 60(11):3534–3546.

Shahabi, C., Ortega, A., and Kolahdouzan, M. R. (2002). A comparison of different haptic compression techniques. In *IEEE International Conference on Multimedia and Expo (ICME '02)*, pages 657–660.

Steinbach, E., Hirche, S., Kammerl, J., Vittorias, I., and Chaudhari, R. (2011). Haptic data compression and communication: A review of the challenges for telepresence and teleaction. *IEEE SIGNAL PROCESSING MAGAZINE*, pages 87–96.