

Toward a Task Space Framework for Gesture Commanded Telemanipulation

Neal Y. Lii, Zhaopeng Chen, Maximo A. Roa, Annika Maier, Benedikt Pleintinger, and Christoph Borst

Abstract—This paper introduces a new framework for task space telemanipulation. The *TASK space grasping and MANipulation (TaskMan)* concept utilizes a library of tasks based on gesture commands, which replaces the conventional mapping required between the human hand and the end effector. Task communication between the human machine interface (HMI) and the robot end effector requires two symbiotic but non-identical state machines on the master and slave side. The task states on two sides are synchronized via a single channel communication, as opposed to multi-channel joint space or Cartesian mapped information. *HAND gesture command for grasping and MANipulation (HandyMan)* HMI command algorithm is proposed for the recognition of hand gestures, which incorporates a library of intuitive task gestures to be used by the teleoperator wearing a CyberGlove. The task gestures are used to drive the states of the TaskMan state machines. With the proposed concepts, this work has realized teleoperated grasp and manipulation with a 15-DoF robot hand in task space. Full 6-DoF of object manipulation was achieved with different grasp combinations, and demonstrated higher repeatability, success rate and easier operation compared to conventional joint space teleoperation methods.

I. INTRODUCTION

Remote operation of a multi-finger robotic hand has been traditionally implemented using bilateral control with visual and force feedback, such that the slave receives commands at the robot hand's joint level. Such telemanipulation requires a mapping between the human hand and a robotic hand, as the kinematics and configuration spaces of both hands are in general different. This mapping is done via the joint space, to get similar poses [1], or in the cartesian space, to get similar fingertip positions [2]. A previous evaluation of the performance of a telemanipulation system for space operation showed acceptable performance in grasping tasks, but poor behavior in fine manipulation [3], mainly due to mapping difficulties, human input retrieval limitations, and end-effector performance limitations. Furthermore, these approaches require large numbers of Degrees of Freedom (DoF) command channels, which would not meet the low bandwidth requirements in some applications such as space telerobotics [4].

Recently, the concept of synergies, i.e. coupled movements of the hand joints, has been presented as a way to overcome the mapping problems. A low number of synergies can cover a large percentage of the variance in human hand configuration and movement [5]. By using a virtual sphere, the human and robot hand synergies can be mapped directly in the task space, thus avoiding the problem of dissimilar kinematics

and dynamics [6]. However, most of the efforts in synergies up to now focus on grasping, whereas investigations into in-hand manipulation are still in the early stages [7].

A proper calibration of the input device at the master side, usually an instrumented glove, is required to guarantee the proper recognition of the human gestures [8], [9]. This gesture recognition is a relevant problem also for the community working on programming by demonstration. Extensive previous work has been done on the classification of human grasp types [10], [11]. However, the problem remains relevant today, as evident by a recent attempt to classify the human manipulation behavior [12].

This paper proposes a task-based approach to robot manipulation. First, the system recognizes how many fingers are suited for a task, and then the task itself is recognized. A free rigid body has six degrees of Freedom (DoF), namely three translational and three rotational. For this work, all six DoF have been successfully realized both at master side (human machine interface, or HMI), and the slave side (high-DoF robot hand). This covers most of the manipulation tasks performed by a human hand to operate most task boards and machineries. In order to extend the capability of the teleoperated robot hand to include tool handling, an additional library of performable manipulation tasks should be introduced as well. For this work, an additional example task type is included for operating pistol grip tools.

The identified tasks are later commanded to the slave. The identification of tasks reduces the bandwidth required in the communication channel, as we are not commanding directly the joint positions, but we rely rather on a suitable control system on the slave side that receives a task command and executes it. This takes advantage of a shared autonomy approach [13].

The rest of the paper is organized as follows. Section II presents the grasp and manipulation gesture recognition, and Section III explains the use of hand gestures to teleoperate a multi-finger hand. Section IV describes the system implementation, and Section V shows some experimental results obtained when applying the approach to the telemanipulation of a five-finger robot hand. Finally, Section VI summarizes the paper and discusses on future works.

II. TASKMAN: TASK BASED GRASP AND MANIPULATION FRAMEWORK

TASK space grasping and MANipulation, or TaskMan, is proposed as a new framework for high-DoF robot end effector grasping and manipulation. Instead of communicating in joint or Cartesian space information, a library of

All authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. {neal.lii}@dlr.de

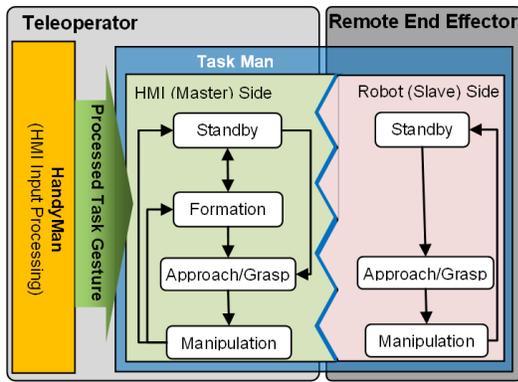


Fig. 1. System view of HandyMan and TaskMan framework. Two symbiotic TaskMan state machines are implemented on the master (HMI) and slave (robot hand) side to communicate and synchronize the current task states. Only the relevant task states need to be implemented for each state machine. As a result, the slave side does not have a formation state. The two TaskMan state machines are linked via an 8-bit UDP channel. HandyMan's task gesture recognition delivers the task gesture commands, which drives the state machine transitions of the HMI side TaskMan. Note that HandyMan and the HMI TaskMan state machine reside on the teleoperator side, whereas the robot (slave) task machine resides on the remote end effector side.

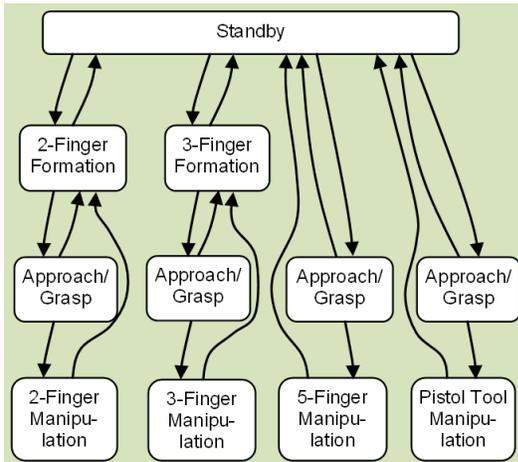


Fig. 2. Detailed view of the master/HMI side TaskMan state machine.

task types (e.g. 3-finger formation/hand preshape, 5-finger grasp, 2 finger X-rotate manipulation, etc.) are employed to realize the operator intended robot hand function. As task space communication is not robot end effector specific (e.g. not bound to any specific kinematics description or D-H parameters), with a sufficiently large task library, it is more easily adaptable to different robot end effector types with different characteristics.

The TaskMan concept for teleoperation is based on dissimilar but symbiotic task state machines, one implemented on the HMI device side (master), and another on the robot end effector side (slave). Fig. 1 illustrates the relationship between the two TaskMan state machines. In the case of a human hand teleoperating a robot hand, the dissimilar state machine reflects the dissimilar mechanics between the operator hand/HMI device, and the robot end effector. As can be seen in Fig. 1, although the two state machines are not identical, they possess similar task states. Instead of utilizing purely quantitative position information to map the operator

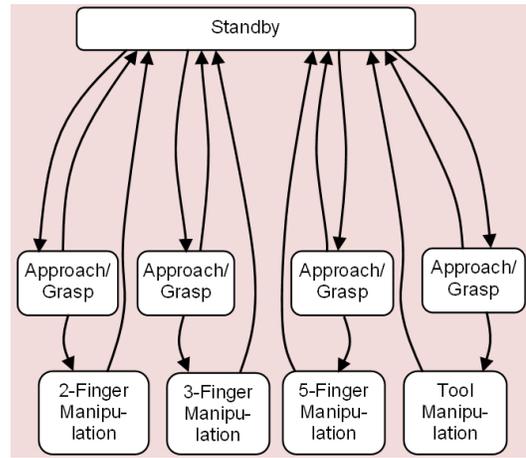


Fig. 3. Detailed view of the slave (robot hand) side TaskMan state machine.

hand as done in some previous work discussed in Section I, the similar task states allow the two sides to synchronize to enable teleoperation.

Fig. 2 and Fig. 3 give more detailed views of the TaskMan state machine for the HMI/master side, and robot end effector/slave side, respectively. The standby, formation, approach/grasp, and manipulation states all correspond to the progression of teleoperation commands from grasping to manipulating an object. They each correspond to task gestures to be discussed in Section III.

The formation command is specific to the HMI TaskMan and not implemented on the end effector side. It serves to narrow down the possible grasp and manipulation type (e.g. 2-finger or 3-finger mode) which simplifies the task gesture recognition, discussed in more detail in Section III. The formation command and task state is transparent to the end effector TaskMan state machine, as it is not directly necessary for the robot hands.

The basis of manipulation is to be able to translate and rotate a grasped object. Consequently, the goal of this work is to achieve rigid object grasp and manipulation in all six DoF with manipulation task states. Additional grasp and manipulation task states for the pistol grip tool, useful in many applications, are also implemented. To realize task space telemanipulation, local manipulation intelligence is required. For this work, 6-DoF of manipulation in 3- and 5-finger modes, 5-DoF of manipulation in 2-finger mode, and pistol grip trigger manipulation are implemented on the end effector/slave side PC.

Currently, the desired approach/grasp position is pre-programmed as a static position due to the lack of an online local grasp planner. However, if necessary, such as when a local grasp planner is not available, and a static grasp formation setting is unsuitable, approach/grasps can be performed in joint space, which exploits the HMI devices strength in grasp performance (but not manipulation, as shown in Section V). The TaskMan concept allows seamless and smooth transition between joint space and task space teleoperation. This is a feature realized by task gestures designed to closely (but not fully) mimic the actual manipulation movement, which

reduces the difference/gap between the task gestures and joint space gestures. The task gestures are discussed in greater detail in Section III. However, for the future, local grasp planning should be added at the end effector/slave side to complete the full task space operation, as a hybrid of joint and task space teleoperation would demand higher communication bandwidth not necessarily available for some applications. This can be accomplished with the addition of local visual servoing and tactile sensing.

As discussed in Section I, communication bandwidth can be a limiting factor in some teleoperated applications such as space telerobotics. The communication between the master and slave side of TaskMan would require only low-bandwidth communication to deliver logic state information to communicate the current intended task state/command. Only a single additional channel is required to deliver the desired displacement of the commanded DoF. This significantly reduces communication traffic as compared to the multi-channel, high-bit displacement commands or measurements required by conventional telemanipulation methods [3].

III. HANDYMAN: HAND GESTURE CONTROLLED TELEOPERATED GRASP AND MANIPULATION

HAND gesture command for grasping and MANipulation, or HandyMan, is proposed to facilitate task gesture commands for task space teleoperation. Previous work has been done on object space hand position mapping for 2-DoF object manipulation [9]. This work has extended the manipulation commands to the full 6-DoF. An initial investigation on task gestures for non-rigid complex objects, such as a pistol grip tool, is also performed. Furthermore, in order to enable grasping and manipulation with different finger configurations, e.g. 2-finger, 3-finger, 5-finger modes, an extended library of gestures must be designed. This work further differentiates from previous work by focusing the task gesture design to fully utilize the sensor implementation of the HMI device. Finally, the HandyMan algorithm for command recognition according to the library of task gestures is described in Section III-B.

A. HandyMan Gesture Library

In order to facilitate task gesture commands for telemanipulation, a library of 43 task gestures has been designed and collected to work with the CyberGlove [14] to achieve 2-finger, 3-finger, 5-finger grasp and manipulation of rigid objects in 6-DoF, as well as a pistol grip tool. 4-finger mode is not implemented as a majority of glove operators encountered during experimentation were unable to retract only the little finger without moving the ring and middle fingers.

The task gestures are designed to maximize the motion capturing capability of the HMI design. Furthermore, they are designed to be intuitive and maintain some resemblance to the actual motions of each task, which enables easier training and operation for the user. This allows the possibility for seamless transition between joint space states (e.g. joint space approach/grasp) and task space states (e.g. task space



Fig. 4. HandyMan gestures: standby.



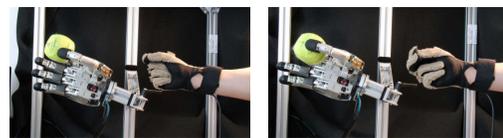
(a) 2-finger formation. (b) 3-finger formation.

Fig. 5. HandyMan gestures: formations.

manipulation). The static and motion gestures are designed so that each can be clearly distinguished from the rest of the gestures in the HandyMan gesture library.

As HandyMan is designed to work with the TaskMan concept discussed in Section II, the task gestures are introduced in terms of their correspondence to TaskMan. The standby gesture, which starts the TaskMan state machine, is designed in the form of a fully opened hand, as shown in Fig. 4. The open hand gesture also signifies the intent of the operator to start or restart grasp and manipulation. For the 2-finger and 3-finger formation gestures, as shown in Fig. 5, the unused operator fingers are retracted to command the intent of 2- and 3-finger grasps that would follow. Note that the robot hand does not retract the unused fingers. This helps to reduce possible obstruction of the hand work space. This would not be possible with conventional HMI mapping methods. Four gestures are designed for the approach/grasp task: three for the 2-, 3-, and 5-finger grasps, and one for the pistol grip grasp, as shown in Fig. 6. Collision detection facilitated by the joint torque sensors makes possible the transition from grasp command to manipulation command without crushing the object or damaging the robot hand. This is achieved by reducing the stiffness once a collision/contact with the object/environment is detected. Once all fingers assigned for the grasping have made contact, the stiffness is increased again in order to grasp the object.

Finally, a total of 34 tasks gestures are designed for 6-DoF manipulation, as shown in Fig. 7 (only the 3-finger gestures

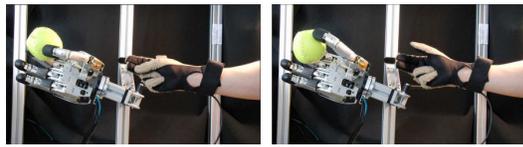


(a) 2-finger approach/grasp. (b) 3-finger approach/grasp.

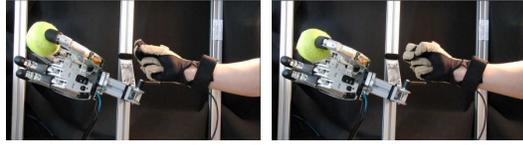


(c) 5-finger approach/grasp. (d) Pistol grip grasp.

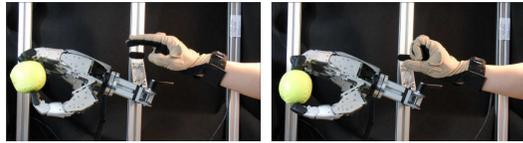
Fig. 6. HandyMan gestures: approach/grasp.



(a) X-rotate manipulation in task space.



(b) X-translate manipulation in task space.



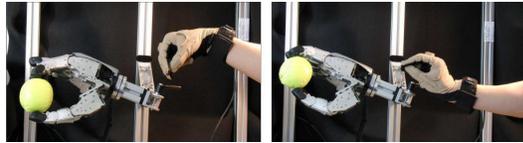
(c) Y-rotate manipulation in task space.



(d) Y-translate manipulation in task space.



(e) Z-rotate manipulation in task space.



(f) Z-translate manipulation in task space.

Fig. 7. Manipulation gestures (in 3-finger grasp mode).



(a) Pistol grip release trigger. (b) Pistol grip squeeze trigger.

Fig. 8. HandyMan gestures: pistol grip tool.

as shown due to space constraint), and two for pistol grip tool operation, as shown in Fig. 8. In our current implementation, the manipulation task gesture coordinate system's orientation is fixed to the robot hand, whereas its origin is set as the centroid of the initial grasp, as shown in Fig. 10.

As can be seen in Fig. 7, the manipulation task gestures abstractly resemble the intended manipulation, but do not exactly mimic these manipulations as they would be performed on a real object. For example, the X-rotate manipulation is completed by simply moving the thumb from side to side. In the case of Y-translate, the human hand is unable to abduct

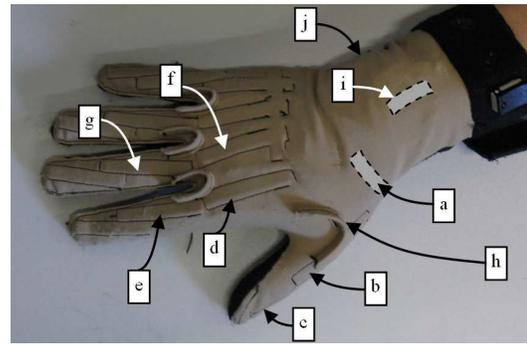


Fig. 9. CyberGlove and the sensors used for HandyMan. The sensor measurements used are: a. thumb rotation, b. thumb Metacarpophalangeal joint, c. thumb Interphalangeal joint, d. index finger Metacarpophalangeal joint, e. index finger Proximal Interphalangeal joint, f. middle finger Metacarpophalangeal joint, g. middle finger Proximal Interphalangeal joint, h. abduction between thumb and index fingers, i. wrist flexion, and j. wrist abduction.

in the Y direction to perform such a manipulation with the fingertips touching (whereas an end effector may be able to). Y-translate is instead achieved by the wrist abduct/adduct motion. Similarly, Z-translate is performed with the wrist flexion motion. This is a (rare) case of the robot end effector actually able to perform a dexterous manipulation that is difficult/impossible for the human hand, at least without physically grasping and manipulating an actual object. The HandyMan task gesture instead incorporates an intuitive alternative gesture which dramatically simplifies the command effort for the teleoperator.

The combination of the HandyMan and TaskMan concepts enables easy adaption to different end effector designs for teleoperation without the need for complicated adjustment and/or calibration to map the operator hand to the end effector. This is obtained thanks to the communication between the HMI (master) and the robot (slave) side consisting of a 'universal' library of tasks, instead of joint or Cartesian measurement/information, which is different from end effector to end effector.

B. HandyMan command processing

As discussed earlier, HandyMan's manipulation task gestures are designed to have distinctly discernible features/movements. This enables the robust recognition of each task gesture type by the HandyMan algorithm. Table I shows the unique motion direction combinations from the observed sensors of the CyberGlove facilitated by the HandyMan task gestures. a. to j. denote the sensors as shown in Fig. 9. A '0' denotes no change in sensor measurement, whereas a green '+' denotes positive rate of change in sensor measurements, and a red '-' denotes negative. The 3-letter DoF in the first column denotes the direction and manipulation type. For example, XRN stands for X-rotation in the negative direction, and YTP for Y-translation in the positive direction.

The 2-finger, 3-finger, and 5-finger manipulation task gestures all make use of the thumb and index fingers similarly. As a result, data in Table I applies to all manipulation categories except for Z-rotate, which can only be achieved in

TABLE I
CYBERGLOVE SENSOR PATTERN/TRENDS OF 6-DOF MANIPULATION
TASK GESTURES.

	a	b	c	d	e	f	g	h	i	j
XRN	+	+	+	0	0	0	0	-	0	0
XRP	-	-	-	0	0	0	0	+	0	0
XTN	-	+	+	-	+	-	+	+	0	0
XTP	+	-	-	+	-	+	-	-	0	0
YRN	+	-	-	-	+	-	+	0	0	0
YRP	-	+	+	+	-	+	-	0	0	0
YTN	0	0	0	0	0	0	0	0	0	+
YTP	0	0	0	0	0	0	0	0	0	-
ZRN	0	0	0	0	0	0	0	0	0	0
ZRP	0	0	0	0	0	+	0	0	0	0
ZTN	0	0	0	0	0	0	0	0	-	0
ZTP	0	0	0	0	0	0	0	0	+	0

3-finger and 5-finger modes. This is due the requirement of a minimum of three fingers to perform a Z-rotate manipulation. In 5-finger manipulation mode, the ring and little fingers follow the same motion as the middle finger, and are not needed for manipulation type recognition.

The manipulation task gesture command algorithm recognizes the manipulation gesture type by utilizing the unique sensor rate/trend combination of Table I with the following weight function:

$$f_w = \sum P_i - \sum N_i - \sum |Z_i| \quad (1)$$

where P_i , N_i and Z_i are the three types of measured rates (positive, negative, and non-moving) observed by each CyberGlove sensor channel. In each data process cycle, the weight function f_w is calculated for all 12 possible manipulation types. The manipulation type with the highest f_w value is deemed the most likely type currently performed. A rate threshold value is used to determine if the weight function output is of sufficient confidence for delivering a manipulation task command. This weighting function is also used for recognizing transitions between different task states such as returning from grasp to open finger formations. In addition, the HandyMan algorithm also employs a series of threshold triggers for the recognition of standby, formation, and approach/grasp task gestures, as well as to reduce the effects of data noise and signal spikes.

Calibration of the CyberGlove for every individual hand requires significant time and effort to fine tune due to the different operator hand sizes, ranges of motion, as well as CyberGlove's sensor variability. HandyMan's rate based gesture recognition method reduces the need for fine calibration for each operator. This is due to the fact that although absolute joint position measurement changes from operator to operator, the intended motion trend corresponds in the same fashion for all HandyMan trained operators.

Finally, upon determining the manipulation type, the commanded translational or rotational manipulation displacement, Δx_{DoF} , is calculated as follows:

$$\Delta x_{DoF} = k_{gain} * (x_{DoFcurrent} - x_{DoF0}) \quad (2)$$

TABLE II
CYBERGLOVE SENSOR MEASUREMENT USED FOR CALCULATING
MANIPULATION x_{DoF} .

DoF	CyberGlove Sensor
X-rotate	Thumb Metacarpophalangeal
X-translate	Thumb Interphalangeal
Y-rotate	Thumb Interphalangeal
Y-translate	Wrist abduction
Z-rotate	Middle finger Proximal Interphalangeal
Z-translate	Wrist flexion
Pistol grip	Index finger Proximal Interphalangeal

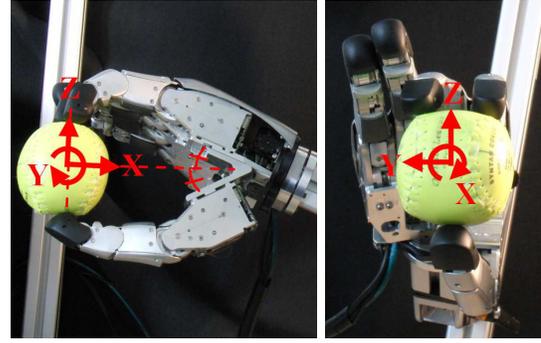


Fig. 10. The modular five finger hand based on components from the DLR/HIT Hand II and its manipulation coordinate system. In this implementation, the coordinate system's orientation is fixed to the robot hand, with the X-axis is aligned to the middle of the angle between the thumb and the opposing finger. The origin of the coordinate system is set to the centroid of the initial grasp.

where x_{DoF0} denotes the CyberGlove sensor value at the time of approach/grasp completion, and $x_{DoFcurrent}$ denotes the current sensor measurement. The sensor measurements utilized for calculating the manipulation amount is shown in Table II. The addition of a gain value, k_{gain} , allows the tuning of gesture command range. High k_{gain} allows for wider range of motion, whereas lower k_{gain} delivers finer manipulation precision.

IV. SYSTEM IMPLEMENTATION

In order to realize TaskMan and HandyMan for teleoperation, a 15-DoF dexterous anthropomorphic hand with five modular fingers based on the DLR/HIT II hand [15][16] is constructed, as shown in Fig. 10. Each identical modular finger has three DoF with a base joint capable of abduction and flexion, as well as a coupled proximal and distal joint. The robot hand model control, task manipulation library, and the TaskMan state machines are realized in Matlab/Simulink. To facilitate impedance control for the robot hand, the controller is implemented on a QNX real-time PC. During operating, a Linux PC can serve as a host to the model running on the QNX machine which allows monitoring of the system operation. However, possibly due to the resource intensiveness of the overall model, long lag can occur in the communication between the host Linux PC and the QNX PC, as can be seen in the attached video. As the QNX PC can function in standalone mode without a host PC, this lag should not pose a real problem for deployment in the field.

The HandyMan algorithm is implemented on a separate Linux PC connected to the CyberGlove to retrieve and

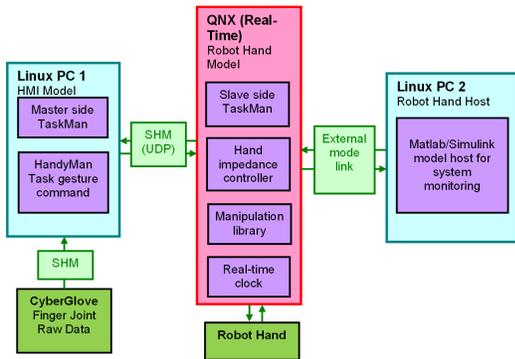


Fig. 11. Implemented system architecture.

process operator commands. In order to synchronize with the real-time environment of the robot hand controller, a real-time clock is implemented on the QNX PC and transmitted to the HandyMan PC. The architecture of the implemented system is shown in Fig. 11.

V. EXPERIMENTAL RESULTS

The experimental results presented here focus on 3-finger grasp-manipulation due to the space constraint of this paper. This configuration is chosen for the reason that three fingers is the minimum number required to achieve 6-DoF manipulation. However, the proposed system is able to perform task space manipulation in any number of finger combinations ($n \geq 2$), provided that one of the fingers is the opposing thumb. The special tool case of pistol grip grasp and manipulation, in the current setup (not discussed in detail in this paper), requires all five fingers.

A. HandyMan task gesture recognition performance

In order to investigate the task gesture command performance, manipulation gestures were carried out in each DoF. The generated x_{DoF} commands are plotted out and shown in Fig. 12. Repetitions of desired commands in the same DoF are performed for each DoF. Repetitions of commands were successfully achieved in all six DoF.

The HandyMan/TaskMan system runs at a 40 msec update cycle, which enables smoothly mixed DoF commands, e.g. mixture of X-translate and Y-rotate in human perceived real-time (this is achieved by recognizing X-translate in one cycle, followed by Y-rotate in the next). However, the fast update rate caused some drift in unintended DoF, which results from bursts of false recognition. The drifts tend to be coupled between certain DoF. One example is X-translate command, which triggers Y-rotate drift in the positive direction, and vice versa. Drift accumulates over continuous series of commands. However, the drift is reset when the operator returns to formation or standby state (e.g. to release the object). Furthermore, the drift can be intuitively compensated by the human operator by commanding in the opposite direction. However, the range of motion in the affected DoF would become increasingly limited as a result, finally rendering that DoF uncommandable if drift persists. HandyMan is nonetheless able to perform several full range

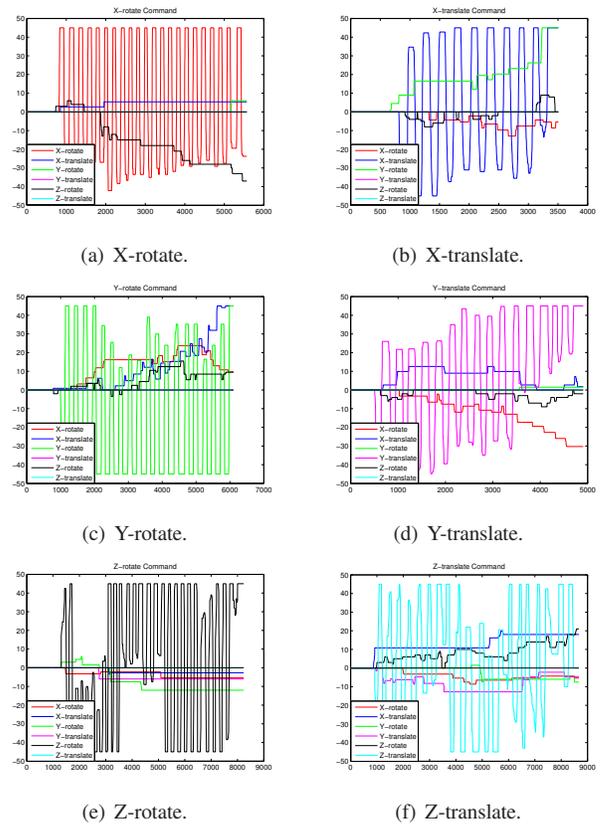


Fig. 12. HandyMan's task gesture command Δx_{DoF} output for task space manipulation. (in 3-finger mode). Drift can be observed in unintended DoF over repetitions of intended DoF manipulation.

motion manipulation commands in the intended DoF before this occurs. As seen in Fig. 12 (b), and (c), in the worst drift case (X-translate command), the Y-rotate drifts fully out of range after 10 full range manipulation cycles in the intended DoF.

The Δx_{DoF} range depends on the position of each finger joint at the end of the approach. Depending on the extension of the corresponding joint, the available range of motion would be larger in one direction (e.g. positive X-rotate direction), than the opposite (e.g. negative X-rotate direction). As a result, the operator should adjust the approach/grasp end position to facilitate more workspace for the desired manipulation task.

B. 6-DoF telemanipulation of an unconstrained object

6-DoF telemanipulation of an unconstrained object was carried out with TaskMan/HandyMan implementation and conventional joint space mapping of the CyberGlove for comparison. Conventional joint space mapping CyberGlove teleoperation command was implemented as in [3]. 20 manipulation attempts were made for each DoF in both task space and conventional mapping mode.

During the task space manipulation experiment, it was first noted that commanding with task motion gestures required some initial period of getting accustomed to. However, once the operator becomes familiar with the manipulation gestures, the operation becomes significantly less straining



(a) Task space telemanipulation. (b) Joint space telemanipulation.

Fig. 13. Task space and joint space (conventional) telemanipulation of an unconstrained object. Two types of command methods can be differentiated by the clearly different positions of the ring and little fingers, which are not used for 3-finger manipulation. Another difference is the end approach/grasp position of the operator fingertips in task space, which are touching, whereas the conventional command shows spacing between manipulating operator fingers.

than that of direct-mapping telemanipulation. With task space manipulation, the operator can focus on delivering a decoupled manipulation command, instead of simultaneously minding the need to keep a sufficiently firm grasp to keep hold of a non-existing object on the operator side while performing telemanipulation commands. Fig. 13(a) shows an example of telemanipulation in task space. It was observed that all six DoF can be operated with high repeatability. Y-rotate manipulation exhibited smaller range of motion than desired. This could be due to the limitations in the manipulation algorithm, which should be improved. Overall, task space manipulation achieved high success rates ranging between 85% and 100% depending on the manipulation DoF, as shown in Table III. With the exception of Z-rotate manipulation, task space manipulation also recorded larger attainable ranges of motion, as shown in Table IV.

In comparison, during conventional joint space manipulation as shown in Fig. 13(b), operation in general ‘felt’ more difficult. This is due to the conflicting performance criteria that the operator must contend with, which calls for the operator to command sufficiently high grasp force to maintain a firm hold of the object, while simultaneously ensuring that the grasp force is low enough to carry out a manipulation command. This would reduce the operator’s ability to perform over a long period of time. The subjective scoring of manipulation difficulty is given in Table IV. One DoF, namely Y-translate, cannot be performed using direct mapping due to the limitations in the human hand. Z-translate was also very difficult to perform. Overall, false manipulation occurred significantly more frequently (e.g. Y-rotate command falsely performed as Z-rotate). Furthermore, the robot fingers frequently lost contact with the object, and the object was frequently dropped. These impressions were confirmed by the compiled experimental data in Table III.

Finally, experiments were carried out successfully to tele-manipulate multiple DoF in series. The ability to perform multiple telemanipulation commands in succession without releasing the object, together with full 6-DoF manipulation capability, means less need to perform a cumbersome sequence tasks of: releasing the object, moving the end effector to a new desired position, and regrasp and manipulate the object again. The release, reposition, regrasp, and manipulate sequence is not only resource intensive, it may also be

TABLE III
EXPERIMENTAL RESULTS COMPARING THE SUCCESS AND FAILURE RATES OF JOINT SPACE AND TASK SPACE OBJECT MANIPULATION.

DoF	Task or joint space	Success rate	Failure rate		
			No movement	False movement	Drop object
XR	Task space	100%	0%	0%	0%
	Joint space	15%	10%	25%	50%
XT	Task space	95%	0%	5%	0%
	Joint space	25%	5%	70%	0%
YR	Task space	90%	0%	10%	0%
	Joint space	40%	30%	30%	0%
YT	Task space	100%	0%	0%	0%
	Joint space	0%	100%	0%	0%
ZR	Task space	85%	0%	15%	0%
	Joint space	80%	0%	20%	0%
ZT	Task space	100%	0%	0%	0%
	Joint space	35%	0%	50%	15%

TABLE IV
EXPERIMENTAL COMPARISON OF THE RANGE OF MOTION AND TASK DIFFICULTY FOR JOINT SPACE AND TASK SPACE OBJECT MANIPULATION.

DoF	Task or Joint space	Maximum attained range of motion	Subjective difficulty of operation
XR	Task space	20°	Easy
	Joint space	3°	Difficult
XT	Task space	17 mm	Easy
	Joint space	14 mm	Difficult
YR	Task space	15°	Easy
	Joint space	11°	Moderate
YT	Task space	26 mm	Easy
	Joint space	0 mm	Impossible
ZR	Task space	31°	Easy
	Joint space	36°	Moderate
ZT	Task space	26 mm	Easy
	Joint space	22 mm	Difficult

unsuitable for some intended manipulation task such as handling fragile objects in confined space.

C. First experiments with 6-DoF telemanipulation of a partially constrained object

An initial attempt was made to manipulate a partially constrained object. A 6-DoF 3Dconnexion SpaceMouse [17] was telemanipulated, which in turn manipulated a virtual box on screen for demonstration, as shown in Fig. 14, as well as in the attached video.

Teleoperation of the SpaceMouse was more difficult than unconstrained objects. This is due to additional necessity to line up the hand’s and the SpaceMouse’s constrained DoF axes. As the axes were better aligned for some DoF

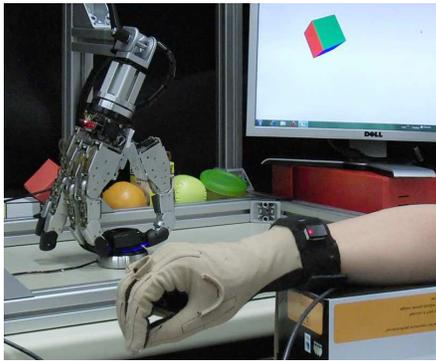


Fig. 14. Telemanipulation of a 6-DoF SpaceMouse in task space. The operator's hand with the CyberGlove is seen at the bottom of the figure, performing a HandyMan manipulation command. The SpaceMouse manipulates, in 6-DoF, a virtual box on screen at the upper right corner.

and worse for others, performances in different DoF were not equal. In comparison, X-rotate was easy to operate and repeatable, whereas Z-rotate and X-translate were deemed the most difficult, and required repeated attempts to achieve successful manipulation.

VI. CONCLUSIONS

This paper presents the concept and implementation of telemanipulation by task space commands. The TaskMan concept is proposed, and realized in the form of non-identical, symbiotic state machines on both the HMI and robot hand sides to facilitate task space teleoperation. Task space commands greatly reduce the necessary communication bandwidth, which is helpful for applications such as space telerobotics. The task space command is enabled by the HandyMan concept, which utilizes a library of task gestures to deliver grasp and manipulation commands from the HMI. HandyMan is primarily based on rate, instead of position measurements, and does not require extensive calibrations. The task space approach also makes it easy to adapt the system to any end effector without complicated remapping necessary with previously known methods. Furthermore, the gesture recognition scheme more effectively exploits the dexterity of the end effector with simple, yet intuitively designed task gesture commands not possible with direct mapping techniques.

The TaskMan and HandyMan concepts were implemented with the CyberGlove HMI and a modular five-finger robot hand. Experiments were carried out manipulating unconstrained and partially constrained objects. Comparisons were made between task space and conventionally mapped HMI driven telemanipulation. The task space concept proved to be superior in terms of achievable manipulation DoF, manipulation success rate, and ease of operation. Task space teleoperation also demonstrated equal or better attainable ranges of motion.

The task space concept introduced here opens up a wide variety of possibilities, as well as many remaining problems to be solved. The task command library should be expanded to include more relevant, non-rigid objects, similar to the pistol grip tool included briefly in this paper. As shown in the

experimental results, the HandyMan task command suffers from command drift. Therefore, more robust task gesture recognition algorithms should be developed. Force feedback should also be investigated to improve user experience and teleoperation performance.

VII. ACKNOWLEDGMENTS

The authors would like to thank Volker Senft for his assistance in setting up the SpaceMouse experiments. Thanks are also due to Peter Meusel and Ralph Bayer for their valuable robot hardware support.

REFERENCES

- [1] M. Ciocarlie and P. Allen, "Hand posture subspaces for dexterous robotic grasping," *Int. J. Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.
- [2] A. Peer, S. Eidenkel, and M. Buss, "Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper," in *IEEE Int. Symp. Robot and Human Interactive Communication*, 2008, pp. 465–470.
- [3] N. Y. Lii, Z. Chen, B. Pleintinger, C. H. Borst, G. Hirzinger, and A. Schiele, "Toward understanding the effects of visual- and force-feedback on robotic hand grasping performance for space teleoperation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 3745–3752.
- [4] G. Hirzinger, K. Landzettel, D. Reintsema, C. Preusche, A. Albu-Schäffer, B. Rebele, and M. Turk, "ROKVISS - robotic component verification on ISS," in *The 8th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space - iSAIRAS*, 2005.
- [5] T. Wimboeck, B. Jahn, and G. Hirzinger, "Synergy level impedance control for multifingered hands," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 973–979.
- [6] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, "Mapping synergies from human to robotic hands with dissimilar kinematics: an object based approach," in *IEEE Int. Conf. on Robotics and Automation, Workshop on Manipulation Under Uncertainty*, 2011.
- [7] N. Hendrich, "Complexities of advanced robotics manipulation (presentation)," *The European Robotics Forum*, 2012.
- [8] M. Fischer, P. van der Smagt, and G. Hirzinger, "Learning techniques in a dataglove based telemanipulation system for the DLR hand," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 1603–1608.
- [9] W. B. Griffin, R. P. Findley, M. L. Turner, and M. R. Cutkosky, "Calibration and mapping of a human hand for dexterous telemanipulation," in *ASME Int. Mech. Eng. Congress and Exposition, Symp. on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2000.
- [10] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Trans. Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.
- [11] T. Feix, R. Pawlik, H.-B. Schmiedmayer, J. Romero, and D. Kragic, "A comprehensive grasp taxonomy," in *Robotics Science and Systems, Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, 2009.
- [12] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *IEEE Int. Conf. on Rehabilitation Robotics*, 2011, pp. 532–537.
- [13] W. B. Griffin, W. R. Provancher, and M. R. Cutkosky, "Feedback strategies for shared control in dexterous telemanipulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 2791–2796.
- [14] Cyberglove systems. [Online]. Available: <http://www.cyberglovesystems.com/>
- [15] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen, "Multisensory five-finger dexterous hand: The DLR/HIT hand II," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3692–3697.
- [16] Z. Chen, N. Y. Lii, T. Wimboeck, S. Fan, and H. Liu, "Experimental evaluation of cartesian and joint impedance control with adaptive friction compensation for the dexterous robot hand DLR-HIT II," *Int. J. of Humanoid Robotics*, vol. 8, no. 4, p. 649, 2011.
- [17] 3D connexions. [Online]. Available: <http://www.3dconnexion.com/products/spacenavigator.html>