21 Teleoperated Service Robots: a new Industrial Revolution

Kees van Hee, Eindhoven University of Technology, k.m.v.hee@tue.nl

Abstract

Teleoperated service robots is a new branche of robotics, different from well-established industry robots and the upcoming humanoids. The main difference is that they are able to perform in principle all tasks a human can do, but not completely autonomous. There is a human operator "in the loop", but this operator does not have to be at the same location as where the robot is performing his task. This means that we are able to decouple the location where an operator resides and the location where his actions are performed. This decoupling could cause a new industrial revolution. In the paper we sketch the main characteristics of teleoperated service robots, the potential economical impact of them and the challenges to realize them.

21.1 Background and Introduction

Why a paper about robots in a liber amicorum for Jo van Nunen? Jo was famous for his work on logistics. His contributions had a wide range: varying from mathematical optimization of logistic systems to re-engineering of supply chains. But do robots fit in this picture? Yes they do! Jo and I used to visit each other several times per year and then during a walk we discussed the "big problems of the world". Of course we often found very innovative solutions, but because we were so "busy", we forgot to tell the world. (Sorry for that.) Often we complained about the academic system, specifically about the way of research funding: big programmes with an appealing name collecting a set of unrelated projects of a group of scientific friends, who would do their own preferred research under the umbrella of the program.me. So we were thinking of a programme as appealing as a-man-on-the-moon, that could be explained in an elevator pitch, but that would contain a set of strongly related research and development projects with high potential economical value. That was a challenging problem specification for a walk in the woods! Our solution of this problem is described in this paper. The elevator-pitch description is:

Make a system of robots that can perform every task a human can do, but controlled by a human at a distance. In this way it is not necessary anymore to move factories to countries with low labour cost, but we only have to move the user interfaces to these countries. So workers in the far east can produce shoes in a factory in the Netherlands, while they stay in their home country.

Such a technology would have revolutionary consequences. To name two: production can be performed at the location of the consumers or where the raw material is produced which saves transportation, and the production technology and the high-tech labour stays in the high income countries, which means that we can preserve the knowledge economy. So this topic fits perfectly in the interest field of Jo. In fact this "great idea" did not disappear and I started a programme for teleoperated service robots (TSR). I have the same background as Jo, we both did a PhD in the field of Markov Decision Processes at Eindhoven University of Technology. Although later I became a professor in computer science, I never did research in robotics. However the problems were so appealing to me that I moved my research interest into this field. Jo joined several meetings at the beginning of the TSR programme and he mentioned the idea in many of his lectures on innovation and logistics. In this paper I will describe what teleoperated service robots (TSR) are, what the economical power of TSR is, how the TSR programme is organized and what challenges there are.

21.2 What are teleoperated service robots?

Telemanipulation is probably the oldest form of robotics. Telemanipulation enables a person, called operator, to act remotely as if the operator was on the spot, by copying the manipulations of the operator at a distance. Telemanipulation is in fact restricted to grasping, moving and releasing physical objects remotely. In this way it is for instance possible to write a letter at a distance by grasping a pen and paper, fixing the paper and moving the pen over the paper. A natural extension of telemanipulation is to transform the human movements, by extending or shrinking the distances, or to increase or decrease the forces executed by the operator. In this way the operator can not only copy his actions at a *distance*, he is also able to act at a different *scale*, e.g. with higher precision, or at larger reach or with more power. Probably the oldest form of telemanipulation is a pantograph, which was used to copy images at a different scale. Typical examples of telemanipulation are found in aerospace and in medical surgery.

Telemanipulation is one of the basic functions of a TSR. But a TSR has more advanced functionality. In order to compete with a human on the spot, the TSR system should be able to perform tasks fast which means that the operator should be able to give a simple command to perform a complex task. In fact we will give our TSR a lot of autonomy in task performance. In the future we will endow the TSR with learning capabilities. Before we define a TSR in more detail, we remark that the TSR field differs fundamentally from the field of *industry robots*. The field of industry robots is mature and the most well-known applications are in the automotive industry. Compared to the TSR the industry robots look much more advanced. The big difference is that industry robots are operating in a completely controlled environment that is often designed for them. To program them, only the *kinematics* of the system are important: the control is completely determined by the coordinates of a position of the robot or its arm. For example the robot arm moves fast (often in an optimal way) exactly to a given position. An industry robot can perform an arbitrary sequence of complex task as many times as we like. But our TSR moves in an unknown an unadapted environment and the operator is not able to give coordinates.

For example consider the movement to a door that must be opened: the operator sees the door via a camera on a screen and he has to give a command to move to the door, to grip the door handle, to move that handle downwards and to pull (or push) the door. This (complex!) task can not be commanded by providing coordinates. There could be obstacles in the room that were not there when a similar task was performed before, so we can not replay the same set of instructions from the past, as is done with industry robots. So industry robots are *numerically* controlled by a *program* for very specific tasks in a completely known environment and they operate *autonomously*, while a TSR acts in a completely unknown environment, on the fly controlled by an operator, who provides *non-numerical* commands. Therefore the field of TSR differs fundamentally from the more classical field of industry robots.

A TSR consists of a *master* and a *slave* component. The slave is in fact the robot that is executing at a distance the commands given by the master. The slave consists of three components: a *mobile platform*, a set of *arms* (one, two or even more) each one equipped with a *gripper* and a *vision system*. The master is an integrated set of devices that enables the operator to control the slave. Seen from a different perspective the master is also a robot, but one with a "human in the loop". In fact, the slave robot is like in telemanipulation, only able to grasp, move and release physical objects.
In a basic TSR the operator has to demonstrate the actions to be executed by the service precisely, may be at a different scale. In advanced TSR the operator has a high-level *command language* in which he can order a complex task for the service robot with a simple command. Such a command can be given to the master by means of advanced input devices such as gloves, joysticks with haptic feedback or by voice recognition. An even more advanced TSR is able to learn behaviour from past behaviour, *programming by example,* and by operator training which is in fact *supervised learning*.
We assume however that our service robot acts in unknown environments, so the system has no map of the environment and there is neither an external navigation system with beacons, nor camera's in the environment that pass information to the TSR. We also do not assume "common knowledge" in our system, so the system does not know that a door turns on hinges and how it should be opened with a door handle. We use the term *service robot* because the TSR replaces the human by performing tasks supposed to be done by humans, sometimes in environments where humans do not like to work, such as in dangerous environments.

There is another class of robots, called *humanoids* that differs from TSR. Humanoids are used to *imitate* humans, while TSR are used to *replace* humans. A humanoid should look like a human and it should be able to imitate emotional contact with humans, e.g. by producing a smile, while TSR only try to take over human actions at a distance and possibly at a different scale, with the least possible effort of the operator.

We expect that TSR need to have some autonomous behaviour for specific domains. On the other hand it seems to be an illusion to be able to program autonomous robots to perform all types of tasks a human can do. One reason for this is that a human can perform many tasks governed by the unconsciousness, like moving the pedals of a bicycle or applauding with two hands. Trying to make this knowledge explicit and transfer it into programs for an autonomous robot, seems to be infeasible. Therefore human control will be necessary for all service robots that act in an unknown environment. Hence we foresee that the gap between autonomous robots (like industry robots and humanoids) and TSR will disappear in the long run: it will always be needed to control robots with a master.

21.3 What is the economical impact?

TSR is a special class of robots, different from industry robots and humanoids which both are autonomous robots. One of the unique features of TSR is that the operator is in the loop which enlarges the scope of applications dramatically compared to autonomous robots. The operator in the loop does not mean that the operator has to be at the same location as the slave. In fact by teleoperations we have *decoupled* the location where a human resides and the location where he performs a task. This is a fundamental forward in technology, comparable with the decoupling of the location where energy is produced and where energy is consumed: electricity is produced in a power plant and we are using it in our homes. This decoupling in energy production and consumption created an industrial revolution. We expect that the decoupling of the location of creating an activity and the location where the activity is performed could create a similar *industrial revolution*.
Another important feature of TSR is acting at a different *scale* as the operator. So at a larger scale the TSR might have longer arms, move faster or it can carry more weight. At a smaller scale the TSR can manipulate objects at the micro level with high precision.
Since a TSR is a system that can in principle act in all situations where a human can act, the TSR are multi-purpose devices. This means that they are in principle suitable for *mass production*, which implies that they can become affordable for a large variety of applications. Typical TSR application domains we can think of, are:

- *Care* for elderly and disabled people. Here we can have two modes of application: the person who needs the care is operating the system himself, or there is another caretaker at a distance who controls the system. Of course it is possible to switch between these two modes. An example of the first mode is some user that is not able to walk and the service robot picks up the news paper from the mail box. An example of the second mode is that a caretaker at a distance helps in waking up a person with a cup of tea.
- *Cure* such as surgery, has two potential modes of use. One is telemedicine, where the surgeon is at a distance, for instance on a cruise ship without a surgeon but with a service robot on board. Also the high precision TSR is important for care, for instance in heart, eye or brain surgery.
- *Maintenance* of equipment or installations requires service engineers to be on the spot. Preventive maintenance can be planned, but in many cases equipment breakdowns require fast repair and for these cases the travelling time to the spot is a bottleneck that can be diminished using TSR. Also maintaining equipment in difficult places, like clean rooms, and in dangerous environments can be improved using TSR.
- *Security* in buildings or compounds is done mainly by means of personal inspection and security cameras. There are great opportunities for TSR here: both to observe objects from unforeseen positions as well as for preventing people to do something or even to arrest them. Also the disarming of explosives is an example in this class of applications. (Today’s robots are only used to inspect explosive bombs.)

- *Manufacturing* is done these days at locations where labour is relatively cheap and not at the location where the final products are needed or where the raw materials are won. With TSR it is in principle possible to have factories "manned" with TSR and operators at a great distance. In a 24 hours economy we could have operators on several locations on earth, each working in its own day time. Also in places where manufacturing is done in an unpleasant environment, we could use TSR, e.g. in a slaughterhouse. A TSR is able to work on a different scale, which might be a specific advantage here.
- *Agriculture* is often very laborious and the margins are so small that guest workers are imported from low-labour-cost countries. In principle it is possible to make a dedicated machine for each kind of harvesting, but economically that is infeasible. With TSR it is possible to let them stay in their own living environment and let them work at a distance, like in manufacturing. Since the TSR are general purpose devices, they can be used for other tasks by other people after the harvesting period. So TSR rental could be a profitable business.
-

Of course there are many more applications, however it is not our intension to give an exhaustive survey. It will be the market that determines which applications of TSR will be successful and which not.

We envision new industrial activities when TSR will really make a breakthrough. First of all there will be companies that produce the hardware components of TSR, such as motion platforms, arms, vision systems, and interface devices for the master. Second a lot of software has to be build and maintained. Third the integration of all components has to be done by system integrators. So they will together produce the complete TSR systems. Then there will arise a new type of *services*, that might be called consultancy. This concerns the deployment of TSR in the various application domains. This requires training of users (operators), adapting business processes and also programming the TSR for specific tasks. Last but not least there will be companies that maintain TSR and companies to let out TSR.

21.4 The TSR programme

Many of the technologies necessary to construct TSR are already available today. So the question is: Why are TSR non existent in daily life? Is it because there is a missing link in the technology or is the integration of the different technologies more difficult than we think or is there no convincing business case?

In our programme which takes two years, we try to answer these questions by building a TSR and to perform experiments . In fact we plan to have three generations of the TSR. The first one, called Rose-0 is operational today (see the photograph below). It is in fact a proof-of-concept and Rose-1 will be the first prototype having all the desired functionality. However with Rose-0 we can perform already experiments and we can test the software that will be used in the later versions as well. The last version Rose-2 will be the production prototype. The project is *experiment-driven*, which means that our specifications and designs are based on use cases and that we test them on prototypes. The prototypes are not only meant to test our drawing table specifications and designs, but they are also used in our *experience* laboratory to learn from experiments (see photograph). The hypothesis is that you only discover problems and their solutions by performing experiments. Actually that hypothesis seems to hold.

In the programme 10 organizations from the Eindhoven region work closely together. The departments of computer science and mechanical engineering of the Eindhoven University of Technology, the technical highschool Fontys, 7 companies with different expertises, in electronics, embedded software, mechanical engineering and one organization from the application domain ZuidZorg, which is the largest organization for home care in the Eindhoven region. We have chosen for this application domain because there is a lot of interest in methods to take good care of elderly or disabled people in a more efficient way.

We try to use as much as possible available components and software. We specifically use open source robot operating system ROS, originally developed at Stanford University and now in hands of the spin-off company Willow Garage (see Willow Garage 2008)). In ROS it is easy to apply algorithms from the Open Computer Vision library see: (Bradiski & Kaehler 2008)) which enables us to reuse advanced technology developed by other teams. For more detailed information and some videos of the programme see (Rose 2010). At the moment the programme is half way.

Figure 1: The proof-of-concept version Rose-0 in the experience laboratory of TU/e with Henk Zeegers, the project manager on the couch and the author standing.

21.5 What are the Challenges?

In this programme there are three kinds of challenges: scientific challenges, engineering challenges and business challenges. Since many techniques and components are available today, one of the main challenges is to combine or integrate them effectively and efficiently in order to reach our goal. Combining existing pieces into a new product is a creative activity. In fact authors arrange existing words in a new way to produce a novel and chemical engineers arrange existing atoms to create new molecules. We emphasize the *scientific* challenges here. For the state-of-the-art of techniques see: (Siciliano & Khatib 2008) or (Siciliano et al 2009). However we discovered by our experiments open problems. We list a few of them.

- *Computer vision.* The operator has to navigate the slave robot through an environment and he has to position the gripper to grasp, shift or release objects. The operator can see by means of cameras on the slave robot where the slave is moving and in particular where the arm with the gripper is moving. It can also see objects in the environment. But the problem is that the camera images are projected on a 2-dimensional screen and so we do not see depth, which is a serious handicap in grasping objects. It is possible to use stereo camera's and to display image such that with special goggles it is possible to see a 3 dimensional picture. However it is not very practical to use these goggles and the precision is also not very good: if you see that the gripper is right on top of the object that should be grasped, you are not sure that this is also the case in reality. There is a huge amount of algorithms for computer vision mainly for autonomous systems. So there are algorithms to make a 3-dimensional computer image of an environment using stereo camera's may be augmented with laser scanners. But these images are not easy to interpret by human operators. So the challenge here is to combine the direct 2-dimensional images with the computed 3-

dimensional images in order to give the operator fast and accurate insight in the position of the slave in its environment. The combination of real and computed images is called *augmented reality*.

- *Simultaneous control*. For industry robots there are all kinds of control algorithms to move to grippers to the right position. But our first problem is that we do not have precise positioning information and in many cases we have feedback via the operator. A more challenging problem is that for certain tasks we need to control both the arm and the mobile platform to perform a certain task. For example for opening a door it could be necessary to move the arm the gripper of which is holding the door handle and simultaneously to move the platform backwards to give the door the space to open. Another example of simultaneous control is when we have to move a plate (for instance with some cups on it) with two hands. Then we do not want to move both arms (grippers) independently, but we want to direct the center of the plate to a particular position and the control algorithms should take care of the fact that the plate moves exactly horizontally in de chosen direction. In both cases we have to combine the control of two different systems simultaneously.
- *Command language and user interaction*. The operator should be able to give his commands in an easy to learn and efficient way. The fist challenge is to determine the right level of commands, which is actually the same as the right level of task complexity and the parameters for the command. The second challenge is to determine the right user interaction device to express the command and its parameters. Here we can profit from the developments in the world of computer games. A particular aspect of the commands is that they have not only input parameters but they may also produce output, which is actually the feedback form the slave robot. Of course we have visual feedback but we can also have *haptic* feedback where the operator can feel forces. We are looking for a good (to avoid the term "optimal") combination of these elements in order to let the TSR be as fast as a human on the spot.
- *Software architecture*. We are developing a *component-based* software architecture that makes it easy to add new functions, even at run time. So when we develop for instance a new method for navigation then it should be so easy to add this function to the system such that it can work in combination with the other functions. So this is the required *plug-and-play* functionality. Further we would like the software to be applicable for different configurations of the slave robot. For instance if we add an extra arm or we replace an arm with a very long or very strong arm then the adaptation of the software should be minimal. In this way we can use the software for a whole family of TSR. So this is the required *configurability* functionality. Last but not least we require that the software architecture enables the formal proof of safety of the robots system. It is of course essential that the TSR will never damage objects or hurt persons when it is in operation.
- *Learning*. We discussed already the need for learning capabilities. For industry robots that is easy: it is just a capture-and-replay functionality. But for TSR this is much more difficult. Consider the task of picking balls form the table and putting them in a basket. The operator is controlling this task with five red balls on the table. What to do if the next time the five balls are in a bit different position? And what to do if there are six red balls or five red ones and a blue one? The big challenge here is not to determine *how* the TSR should learn but *what* exactly should be learnt.
- *Object recognition*. It must be possible to give the TSR the command to find a coffee cup. This means that it should be able to recognize a shape that fits the characteristics of a coffee cup. There is a lot of research in this area today, but it will take some time to let the results be useful in the TSR. We may expect that in the future most daily life objects will have a 3-dimensional image in a public library. Since most objects are designed with CAD (Computer Aided Design) systems, the data are already available.

There are all kinds of *engineering* challenges. One of them is the *speed* and *reliability* of the communication between the master and the slave. Another one is to design the slave and the master in such a way that it is relatively easy to make a *family* of TSR with different scale properties (bigger, stronger, smaller, more precise). This requires a component structure that allows to assemble different variants of components in an easy way. Of course the greatest challenge is to make the TSR for a reasonable price!

There are also *business* challenges. In the Eindhoven region there are many high-tech companies: some big OEM (Original Equipment Manufacturer) companies and many small suppliers.

For the big OEM companies TSR is not a strategic product (yet). The high-tech suppliers are very well equipped to build parts of the TSR, but they lag an entrepreneurial culture: they always wait for a detailed order from an OEM company. So a nice solution could be to set up a network organization where each supplier is a node and where one node is added: the sales node.

21.6 Conclusion

We have sketched the development of a new type of robots, TSR and their great economical potentials. We enumerated a set of challenges connected with the development of these robots. And we sketched a programme focussed at the development of TSR. Jo has actively participated in the conception of the programme. Jo was a very dear friend and I am very grateful for his friendship in general and in this project in particular.

References

Bradiski, G., Kaehler, A. (2008) Learning OpenCV, Computervision with the OpenCVLibrary. Riley Media

Siciliano, B., Khatib,. O. (2008) Handbook of Robotics. Springer-Verlag Heidelberg. ISBN: 978-3-540-30301-5

Siciliano, B., Sciaricco, L., Vilani, L. Oriolo, G. (2009) Robotics: Modelling, Planning and Control . Springer-Verlag London. ISBN: 978-1-84628-641-4

Rose (2010), http://www.robot-rose.nl

Willow Garage (2008), http://www.willowgarage.org