Building Local Maps in Surgical Robotics

Philipp J. STOLKA and Dominik HENRICH
Lehrstuhl für Angewandte Informatik III
(Robotik und Eingebettete Systeme)
Universität Bayreuth, D-95447 Bayreuth, Germany
{philipp.stolka, dominik.henrich}@uni-bayreuth.de

***Abstract* - Currently, robotic systems employ almost exclusively global sensor information for navigation purposes. While a global map facilitates planning, it may have insufficient quality. Especially with autonomous robots, additional information – intra-process, spatial, current, and persistent sensor data – proves necessary to cope with uncertainty, measurement errors, and incompleteness of data.**
We aim at augmenting the world model for standard robotic systems by implementing local navigation and map-building based on local sensors (which provide local data without primary location information), with the use of force and audio classification in a medical robot system (RONAF) in orthopedics applications as an example.

Index Terms – local navigation, local sensors, map-building, classification, medical robotics

I. INTRODUCTION

Navigation in general consists of acquisition of data from the environment, map building, maintenance of the map, and planning of subsequent motions based on this map. The problems in performing these subtasks are similar for both technical and non-technical systems: to gather information, to represent the environment in a manner suitable for building and reading the map, and to plan motions according to a superordinate goal.

One can clearly differentiate two kinds of situations: the map can be available before process execution starts, or it has to be built, completed or maintained during execution. Furthermore, two conceptually different types of sensors can be used: *global sensors* which collect data from a large area and return it with associated position information (the data is embedded into a coordinate system, e.g. in radar imaging), and *local sensors* that are more restrained in their range – they collect only data corresponding to the immediate neighbourhood of the sensor and without an associated coordinate system (e.g. a pressure sensor).

Currently, stationary robots – including both industrial and medical robotic systems – and to some extent mobile robots employ almost exclusively maps from global sensors for navigation purposes. These maps can be built from radar, laser, or ultrasound imaging (US), from (stereo) cameras or camera arrays, from computer tomography (CT) or magnetic resonance tomography (MRT), and many other sources. While a global map facilitates the planning process, it may have insufficient quality in terms of precision, timeliness etc. Although there exists a large body of works concerning the *simultaneous localization and map-building* (SLAM) problem for mobile robots, the case is different for stationary robots. We aim at augmenting the world model for these systems by implementing map-building and navigation based on local sensors, with the case of the medical robot system RONAF [3], [7] as an example for orthopedic applications (Figure 1).

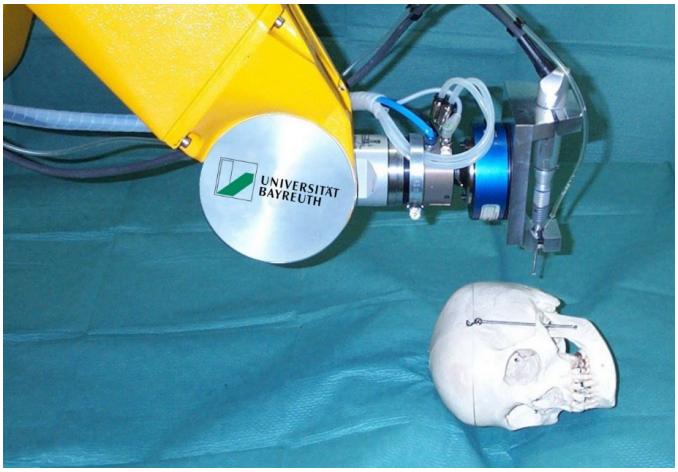

Figure 1: The RONAF system. From left to right: robot, pneumatic overload protection device, force-torque sensor, surgical milling tool, and skull

In Section II, we give an overview over the state of the art in local sensing in stationary robotics. Section III provides a classification of useful navigation principles to be found in robotics. The local sensors used in the presented system are introduced in Section IV. Section V describes the local map data structure and related considerations. The final goal of navigation, the modification of motion paths, is briefly illustrated in Section VI. Finally, Section VII gives an outlook over our current work in this area.

II. STATE OF THE ART

Nowadays, stationary robotic systems widely and almost exclusively employ global sensors for navigation. This includes imaging sensors (CT, MRI, US) for preoperative planning and confirmation purposes as well as positioning sensors (encoders, trackers) for intraoperative use.

Local sensors, in contrast, are rarely used, e.g. force/torque sensors or proximity detectors which operate in control cycles. Prominent examples are the control of feed rate in milling applications based on force measurements [2], [2], [4] or tool deflection correction [2]. Another sensor development in the presented system is a highly precise A-

mode ultrasound scanner for thickness measurements [5], which can be used for bone volume reconstructions in conjunction with tracking. Laser surface scans and fluorescence measurements for tissue discrimination are other possibilities under early development [9].

In milling applications, the contact between miller and material provides a wealth of visual, haptic, tactile, and olfactory clues to the current situation. However, they receive comparatively little attention. One area of work is using force data to classify certain aspects, e.g. the classification of surgeon skill based on force measurements at endoscopic grippers [11], where Hidden Markov Models prove useful to analyze the low-frequency flow of data. Another area under development is the high-frequency analysis of vibration or sound [1], where microphones record audio samples during bone milling which are classified into bone contact states through neural networks, support vector machines or Hidden Markov Models.

No approaches seem to exist which aim at navigation on data sampled through local sensors, and map-building is considered only insofar as tracked US or laser scans are used for reference purposes or registration control.

III. Navigation Principles In Medical Robotics

Current surgical robot systems mainly rely on two sources of information: global spatial data sampled during a planning phase before process execution (*preoperatively*), which is then used statically for global navigation, and local data sampled during the process (*intraoperatively*), which is fed back and used in a non-spatial context in open- or closed-loop controllers of the process. The former may lack either resolution, segmentability, or both, while the latter only persists during the instant of sampling and is discarded immediately after entering into the control cycles. However, especially with autonomous robots, additional information – intraoperative, spatial, current, and persistent sensor data – proves necessary to cope with uncertainty, measurement errors, and incompleteness of data.

The following Subsections A through D describe four navigation principles which can be encountered in medical robotics, while Subsection E incorporates all in one navigation system architecture (Figure 2) [8].

A. *Global Navigation, Preoperative Data*

Global navigation with a preoperative map requires preoperative imaging of the intervention region and is used mainly for planning. Locations and paths can be described within this map in a global fashion. This map needs to be registered with the environment before process execution. No strict temporal restrictions between data sampling and process execution are imposed.

B. *Global Navigation, Intraoperative Data*

Global navigation based on an intraoperatively acquired map is conceptually similar to the previous principle. One has knowledge of the complete environment via a global map. However, acquisition may take place occasionally during the intervention, thus updating the environment representation. Moreover, co-registration, i.e. combining data sampling with localization of the robot, is possible in this case.

C. *Local Navigation*

In contrast to the two principles above, local navigation does not require an environment map before process start. Execution begins without prior knowledge, and a local map is continuously filled with information sampled during execution. The new information has two important properties: it is local, and it may provide more current and precise knowledge of the environment than global sensors could, depending on the technology used. If the position of the sensors relative to the robot is known, this map and the robot are implicitly registered. The information is sampled in tight temporal relation to the process, so it is as up-to-date as possible.

D. *Control*

Control encompasses the measurement of data elements from the process, computing a reaction in a controller that is fed to an actuating element, and possibly providing a feedback path to the controller for closed-loop control. Control as such does not require spatial information; it serves as a reactive navigation principle without persistent map-building functionality.

E. *General Navigation System Architecture*

Almost all surgical procedures are subdivided into two main phases, one for the preparation (*preoperative phase*, e.g. implantation planning) and one for the execution of the surgical intervention (*intraoperative phase*). The mentioned navigation principles can be described as four sensory feedback cycles over both phases (Figure 2).

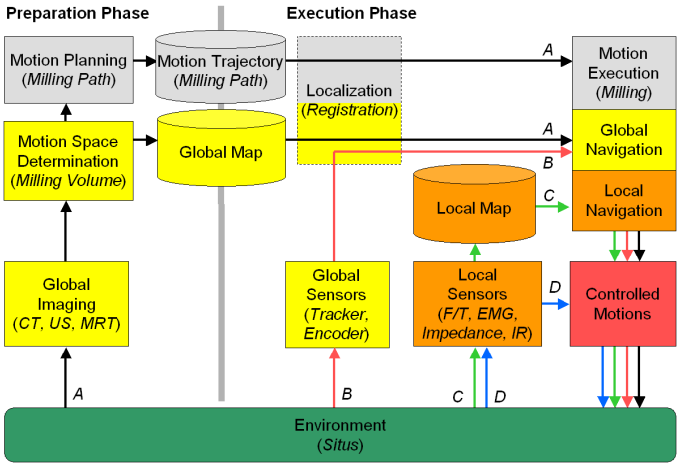

Figure 2: General navigation system architecture including the four navigation principles (A through D), with examples from surgery in brackets

IV. Using Local Sensor Data

We implement the presented navigation principles in the medical robotic system RONAF ("Robot-based navigation for milling at the lateral skull base") [7]. The scope of application is the automated milling of cavities in skull bone for subdermal

implantation of hearing aids. One step of the process is the removal of bone material from the thin calotte in the shape of flat amplifier components, with the possible complication of breakthrough into the skull interior and the rupture of the sensitive meningeal (*dura mater*) enclosing the brain. No exact value for maximum allowable force exerted by the rotating mill on the dura mater can be given, since the rupture threshold is highly dependent on parameters like rotation speed, angle of attack, and wear of the mill head. Though earlier experiments have shown that comparatively high forces (in the range of 20…30N) can be sustained without ruptures, this is only valid for perpendicular orientation of dura and miller head, which is impractical when flexible handling is demanded. We aim at enhancing safety by performing automatic sensor-controlled milling.

In the following Subsection A, we describe the general flow of control in the presented system, while Subsections B and C elaborate on details of the two employed local sensors, force and audio sensing.

A. System Architecture and Flow of Control

According to the general scheme described in Section III, the RONAF system uses the global sensors computer tomography (CT) and robot joint encoders. Local sensors yield force/torque data (F/T) and audio data (through a room microphone).

In the *preparation phase*, the planning system reads the volumetric descriptions of the milling volume (CAD geometry of a hearing aid implant) and the milling material (global CT scan of the region of interest in the skull). It cooperatively computes an optimal position for the volume within the material by minimizing the mean square distance of the upper profiles of both, while complying with certain other constraints [13]. The intersection volume is computed, and completely-covering milling paths are determined within this effective milling volume according to parameters like milling tool tip shape or surface finish [14]. The motion trajectory is then transferred to the robot controller for execution.

In the *execution phase*, first the robot localizes itself (i.e. registration is established between the motion trajectory and the actual environment). Currently, this task is performed non-invasively by guiding the tool tip manually to three surface points defining a local coordinate system whose relative position is determined through joint encoder readings [12]. Then, milling takes place within this local coordinate system. Constant monitoring with local sensors continually updates a system state s_{RFC}, consisting of

- *R – robot motion state* (0: standstill; 1: moving with a speed v_{rob} higher than an application-specific minimum speed v_{min}),
- *F – miller operational state* (0: stopped; 1: rotating),
- *C – contact state* of miller and environment (0: no contact/air; B: bone contact; D: dura mater contact; R: resonance of the miller shaft).

The resulting system states are listed in Table 1. They are discoverable through different logical sensors or classifiers. The robot motion substate *R* is classifiable through direct measurement of the differentials of position over time. The states marked *f* can be detected by force measurements (Subsection IV.B), those marked *a* by audio sampling. The hatched states $s_{00\mathrm{R}}$ and $s_{10\mathrm{R}}$ are impossible, as there can be no miller resonance without miller operation. The states $s_{00\mathrm{B}}$ and $s_{00\mathrm{D}}$ (complete standstill on bone or dura) are not only indistinguishable with the presented sensors, but also improbable to occur except in case of complete system failure, which needs to be recovered through the human operator, anyway. The states $s_{10\mathrm{B}}$ and $s_{10\mathrm{D}}$ (motion under contact without miller operation) lead to exceedingly high forces which cause system shutdown through several cascaded safety circuits [3]. All other states are covered by one or more classifiers.

TABLE 1: LISTING OF ALL SYSTEM STATES s_{RFC}. CLASSIFIABLE STATES ARE MARKED, *f*: BY FORCE SENSOR, *a*: BY AUDIO SENSOR. HATCHED STATES ARE IRRELEVANT/IMPOSSIBLE.

	s_{RFC}	*R* = 0		*R* = 1	
C	0	000 *f, a* 000	010 *f, a* *10	110 *f, a* *10	100 *f, a* 100
	B	00B — (indist.)	01B *a, f* *1B	11B *a, f* *1B	10B — !!!
	D	00D — (indist.)	01D *f* *1D	11D *f* *1D	10D — !!!
	R	00R	01R *a* *1R	11R *a* *1R	10R
		0	1		0
		F			

For every milling intervention plan, there exists an expected state sequence which can be monitored through local sensors. The system states are used for controlling (error detection) and navigation as described in Sections V and VI.

B. Force Sensor Data

The 6D F/T sensor (JR3 90M31A using strain gage bridges, 63N/5Nm max. sensing range, resolution 1:4000) is mounted between the robot tool flange and the milling tool (Aesculap microtron EC/GD622 electrically driven miller with up to 30.000 rpm; Figure 1). It serves as a monitoring device for the low frequency system response (i.e. forces). Force data vectors $\mathbf{F}(t_i)$ are sampled continuously at a varying rate of 25…200Hz. A feature vector $\mathbf{v}(t_i) = (\mu, \sigma)_{|\mathbf{F}(ti)|}$ of the sliding average and standard deviation of absolute forces $|\mathbf{F}(t_j)|$ over a period of d = 0.1s is computed, and a Bayes classifier assigns system states s_{RFC}. These features were chosen because of their relative invariance to "irrelevant" transformations (effects like changes in direction of milling motion or sample rate). Furthermore, they can be efficiently determined even on slow computers. (Note that we do not assume any special distribution of the samples as of now.) Meanwhile, the robot

speed is controlled by virtue of force-based speed control [4], thus keeping the effective absolute forces within certain bounds (set at F_{target} = 20N for milling in bone).

In principle, this classifier can be sensibly used to determine the states of the F and C variables. For training, the force data stream of a complete milling intervention was recorded and manually segmented. The feature vectors were computed and their values discretized (at steps of $\Delta\mu = \Delta\sigma =$ 0.2N within the range of 0…5N) to fill non-parametrical class-(state-) conditional density function matrices $p(\mathbf{v}|s_{RFC})$, as seen in Figure 3 and Figure 4 for classes s_{000}, s_{010}, s_{100}, and s_{110} (varying R and F under C = 0, respectively). As the progress of milling is unknown from the viewpoint of the classifier, the a-priori probabilities $P_{RFC} = P(s_{RFC})$ are set to the same value for all states s_{RFC}. During later execution, the a-priori probabilities for s_{1**} and s_{0**} (robot moving/standing still) are set to 1.0 or 0.0 according to the result of the external robot motion indicator. We apply the general error-rate-minimizing Bayes decision rule of deciding for maximum posterior $P(s_{RFC}|\mathbf{v})$. Classification then takes place every Δt_{aa} = 0.05s during execution.

We extended this force-based classifier to detect different contact states C, especially for differentiation between C = B and C = D. As C = R is characterized by high frequency components, we cannot hope to distinguish it with force measurements in this system. Figure 5 shows the histograms for states s_{11B} and s_{11D}; those have been determined with a slightly differing system setup (without network transmission).

C. *Audio Sensor Data*

Apart from using the *in-situ* available force data, experience hints at another source of information about the system state: recorded sound from the process. While forces can be measured only through a low-pass filter inherent to the sensor system, the audio channel serves as a complementary high-frequency data source (comparable to measuring vibration data, only without spatial direction). Due to the high sample rate and computation load, processing and classification takes place on a personal computer (PC) – either the process control PC or a separate computer. In the RONAF system, audio data is sampled through a directly attached room microphone mounted on a tripod stand (about 1m above and to the side of the region of intervention, obstacle-free), processed, and classification results with timestamps are transmitted to the main control system over TCP/IP (Transfer Control Protocol packets over Internet Protocol via Ethernet). This allows synchronization of data with those from other sources that carry attached spatial information (e.g. contact information as described in Subsection IV.B), which is necessary to "localize" the audio information.

In the RONAF system, samples $s_{audio}(t_i)$ are continually recorded at 44.1kHz (mono) with 16 bit resolution during the whole process. Each real-valued 512-sample block $S_{audio}(t_i)$ (corresponding to ~11msec or ~86Hz) is subjected to a real Fast Fourier Transformation (FFT) with Blackman windowing [6], directly resulting in a 257-dimensional real feature vector

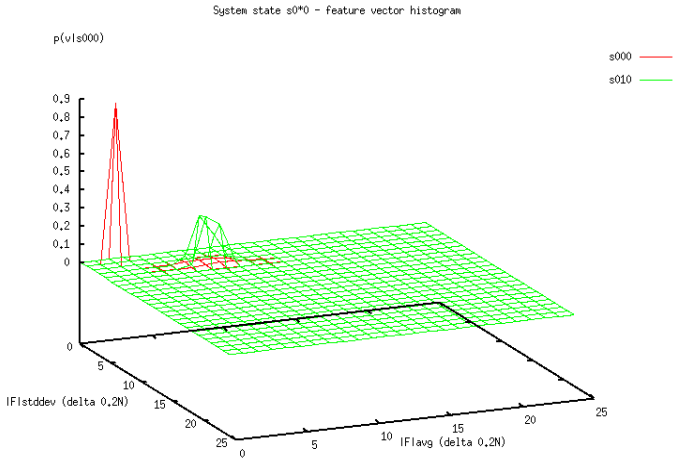

Figure 3: Feature vector histogram for s_{0*0} – robot in standstill, miller stopped/rotating

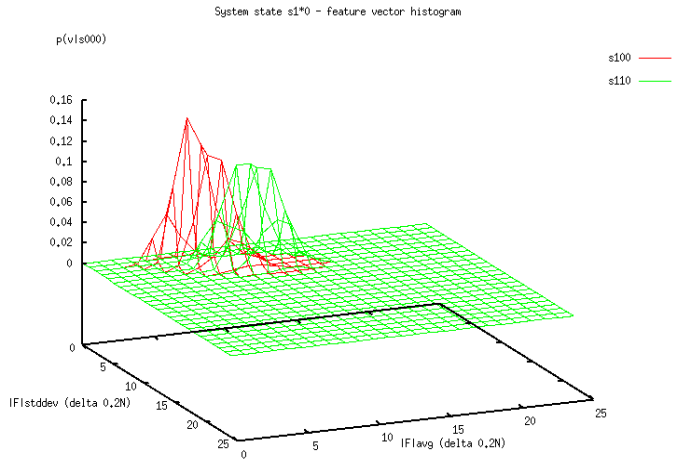

Figure 4: Feature vector histogram for s_{1*0} – robot moving, miller

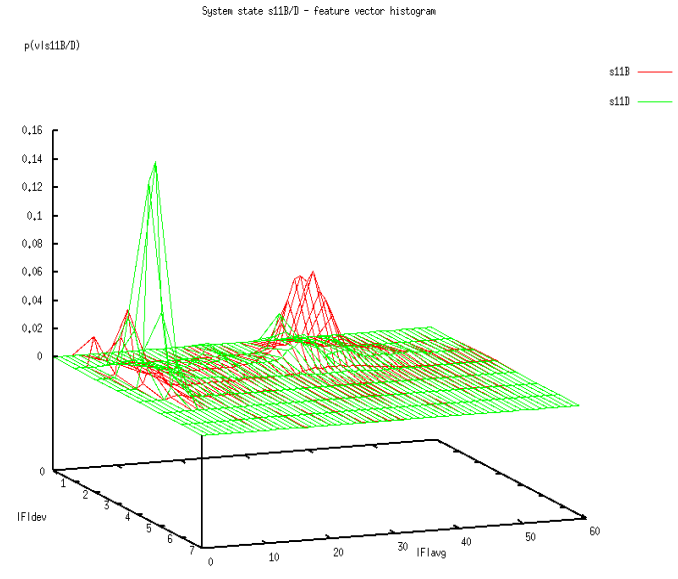

Figure 5: Feature vector histogram for $s_{11\{B,D\}}$ – robot moving, miller rotating, contact with bone or dura

$\mathbf{v}_{FFT}$. (The dimension is due to the fact that the n input values are all purely real, and therefore the transform is symmetric with $\lfloor n/2 \rfloor$ non-redundant values plus one "DC" offset value. Also note that we do not assume any significant second-level features calculated from the FFT, so classification is done on the $\mathbf{v}_{FFT}$ themselves.) As creating reliable, completely-covering histograms for the class-conditional probabilities $p(\mathbf{v}_{FFT}|s_{RFC})$ in such high-dimensional feature spaces is nearly impossible – at least with limited training sets – we use statistical clustering with a one-layer perceptron neural net (Nearest-Neighbour approach) to classify each single $\mathbf{v}_{FFT}$ [10]. We aim at differentiating between the classes s_{000}, s_{100} (i.e. robot standing still without milling and moving without milling), s_{*10}, s_{*1B}, and s_{*1R} (milling without contact, under bone contact, and with resonance). As will be seen, robot motion R is not determinable with audio sampling.

To this end, we recorded a complete milling intervention and manually segmented the audio stream into clearly distinct process phases. The stream (classified, with attached s_{RFC} labels) has been fed through a FFT algorithm and the resulting feature vectors $\mathbf{v}_{FFT,i}$ were alternately assigned half to a training set V_{FFT_train}, half to a test set V_{FFT_test} (of about 4760 vectors each). For training, the class labels of training vectors were discarded, and a set N of m neurons n_i was initialized with weights $\mathbf{w}_i = \mathbf{v}_{FFT,j}$ (random samples in V_{FFT_train}). As the distribution of feature vectors was unknown, the number m of neurons had to be chosen large enough to ensure that all possible sub-clusters are covered. (For the mentioned five classes, $m = 50$ proved later to be satisfying.) Now, all samples $\mathbf{v}_{FFT,i}$ from V_{FFT_train} were presented once in random order, adjusting the nearest neuron towards $\mathbf{v}_{FFT,i}$ according to the Euclidean distance (*learning phase*). After this step, samples were presented to the net again to determine the represented class $s_{RFC}(n_i)$ of every neuron n_i by creating class activation histograms $H_i(s_{RFC})$ over input classes for each neuron and taking their respective maximums (*association phase*).

TABLE 2: RESULTS OF NUMERICAL AUDIO CLASSIFICATION

		System State – Recognized				
		s_{000}	s_{100}	s_{*10}	s_{*1B}	s_{*1R}
System State – Actual	s_{000}	0,757	0,243	0	0	0
	s_{100}	0,573	0,427	0	0	0
	s_{*10}	0,048	0,174	0,775	0,003	0
	s_{*1B}	0	0,002	0,060	0,938	0,001
	s_{*1R}	0	0	0	0,452	0,548

Testing classification rates against the test set V_{FFT_test} yielded results as shown in Table 2. Except for samples from s_{100}, all other states are mostly classified correctly. As there occurs a mix-up between s_{000} and s_{100} judging only from audio data, we cannot reliably distinguish between a standstill and motion of the robot (noise obscures the robot hum). However, robot motion R can be deduced from other sources, as described before. Milling in air ($C = 0$) and in bone ($C = B$) can be distinguished clearly, while resonance ($C = R$) is only mixed up with bone milling, which is a closely related system state. Conclusively, we can detect the operational state F of the miller and the contact state C with a high sample rate (~86Hz).

V. MAP-BUILDING AND READING

Using the state information from force and audio classification, we can instantaneously supervise the system w.r.t. certain safety aspects. However, local sensor data exhibits much greater utility when it is associated with spatial information, i.e. its place of origin. If the position $\mathbf{p}(t_i)$ of the local sensor probe is known, the temporal stream of local sensor data $s_K(t_i)$ can be mapped into a new representation of the environment. (The term *sample* or *datum* $s_K(t_i)$ shall denote a result from one of the previously discussed classifiers K at time t_i, unless noted otherwise.) Depending on the nature of the corresponding local sensor, this new local map can enhance the completeness, correctness, precision, and/or timeliness of global maps available before process execution. Besides, it can serve as a means of process documentation for later reference. Another use of a local map is controlling the registration quality, i.e. concurrently comparing actual locations of environment features with those expected from an initial registration of the robot and available maps.

A. Map Design Considerations

Other considerations concern the map data structure. There is a wide range of possibilities for the organization of data samples. One extreme option would be continuous organization, i.e. saving samples $s(t_i)$ together with their associated position $\mathbf{p}(t_i)$ directly into a linear list $L = [s(t_i), \mathbf{p}(t_i)]_i$, with the advantage of the maximum possible reconstruction resolution along the path when reading the map. However, there is no efficient way to access the data other than according to a strictly temporal order, rendering this approach expensive for spatial access to long lists (i.e. mean complexity between $O(N/2)$ and $O(N)$ for position $\mathbf{p}$ as search key, depending on the relative probability of requesting either existing or non-existing samples, i.e. the frequency of searching through the whole list to its end; with N being the sample count). A solution to this problem is the use of an additional grid superimposed on the list, where the grid entries serve as a hashing structure for faster spatial access, while preserving the resolution advantage of the first approach (mean $O(N/(2n))$ to $O(N/n)$, respectively; n being the grid element count). The relative gain of this method obviously depends on the sample rate, the grid resolution, and the memory requirements of storing the location information, and therefore needs to be evaluated on a case-to-case application-specific basis. When the absolute location information requirement can be loosened and the location $\mathbf{p}(t_i)$ discarded, a sufficiently high-resolution grid can be used, thus discretizing the map into voxels (buckets) with spatial access complexity $O(1)$. Finally, the other extreme would be using one single voxel which collects all $s(t_i)$ without associated positions, allowing only statistical reasoning over the map without spatial reconstruction. Depending on speed requirements during execution (if any), there exist time constraints for the (mostly spatial) access to single data elements, which may be relevant

when e.g. a complete iteration of the sequential map list would be necessary to determine neighbourhood properties. Conversely, a grid representation is likely to have higher space requirements, even though most voxels will be empty for high resolutions and quasi-random traversal of the map by the map-building local sensor.

Closely related with this choice of map structure is the mode of sensor data acquisition – whether the sensors *push* the data into the map according to availability, which sorts the data into voxels, if necessary, or whether the map itself *pulls* the data from the sensors when a trigger rule fires, e.g. upon entry into a new voxel or after a timeout.

Therefore, the map-building problem can be expressed in terms of structuring a map M and defining two functions, $b_M: C \times t \times \Re^3 \rightarrow M$ for map-building and $q_M: \Re^3 \times t \rightarrow C$ for querying the map.

B. *The RONAF Map Structure*

Based on the above-mentioned considerations, the local map structure in the RONAF system is implemented according to the general principle in Figure 7. As the local sensor information shall enhance the available global map data, which originate from rastered imaging sources, we decided on a gridded 3D representation M of the workspace. The resolution (size of the single voxels m_p at position $\mathbf{p}$) is chosen high enough to allow discarding the sample location information without significant loss of precision; in our case we chose an isometric resolution of $\Delta x = \Delta y = \Delta z = 0.35$mm. (This equals to both the relative accuracy of the robot and, in turn, the resolution of the global map used for motion planning before execution.) To take advantage of high sensor sample rates and to minimize the effect of misclassifications, all data is pushed into the map upon availability. Furthermore, this frees the map administration module from having to anticipate the robot motion on-line. (Instead of following the motion plan, the controller might direct the robot into pause positions etc. An algorithm which intelligently pulls data into the map would be considerably more complicated if it had to deal with such situations.)

The issues relating to these decisions can be summarized as follows:

- Jitter of the sample rate is present, i.e. the time span Δt between samples varies.
- As a consequence, the number n_p of samples at grid position $\mathbf{p}$ varies – it may range from zero over one up to "many" samples per voxel, as the data rate is comparatively high. At a motion speed of $v = 5$mm/s and a sample rate of 86Hz, covering 0.35mm creates already six samples.
- As a single voxel m_p can be visited several times, e.g. during tool return motions, we need to differentiate between different temporal "batches" of samples within one voxel. The number of traversals through one voxel ranges from zero over one to "many".
- Finally, we require spatial and temporal reconstruction methods. This includes both sample insertion functions into the map as well as sample retrieval functions from the map.

The map-building function $b_M(s_K(t_i),t_i,\mathbf{p})$ performs

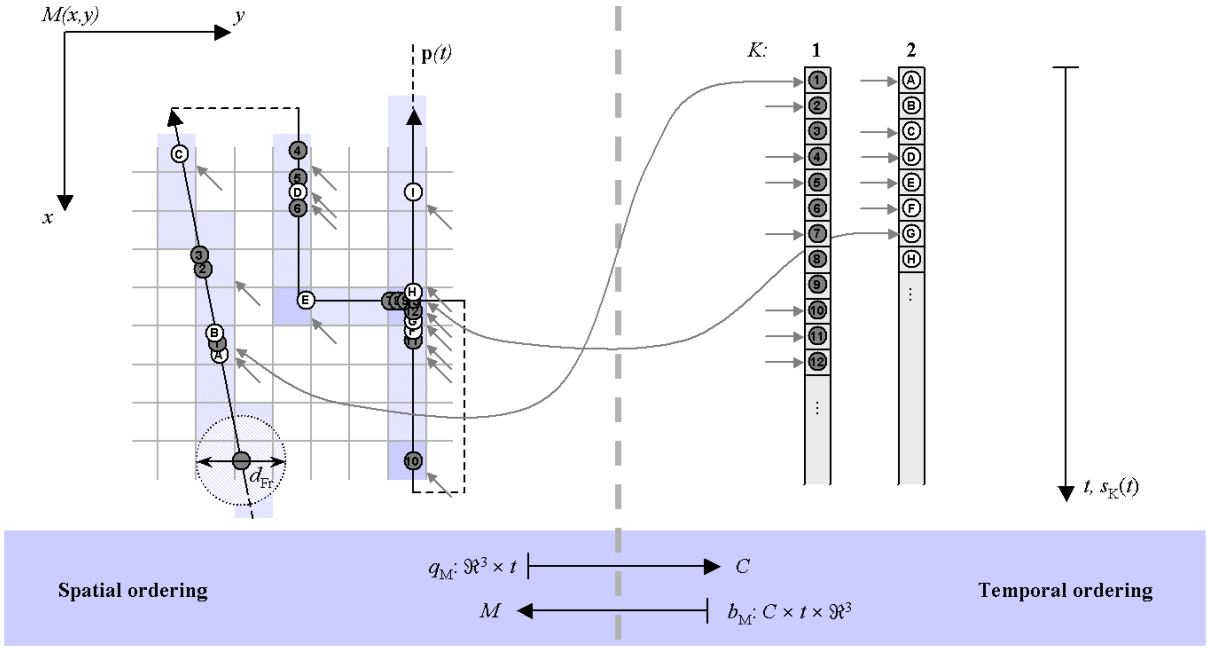

Figure 7: Left: Voxel grid of spatial map M with tool tip path $\mathbf{p}(t)$ (long black arrows); right: temporal sequence of acquired samples $s_K(t)$ from classifiers K_1, K_2. Sample locations along path and time are shown as small circles. Voxels $m_p \in M$ can point to the beginning of their associated sublists in the sample lists, depending on the existence of such samples. Each sample list is in chronological order. Its composing sublists point to the respective voxels in turn. (Each arrow pair – only two of which are shown – between the spatial and the temporal side represents such a bi-directional association.)

sequential insertion of samples $s_K(t_i)$ into separate lists for each classifier data stream K (force, audio, …) in each voxel m_p. A new collection S_{tv} of data-stream-specific sublists is instantiated at the time t_v of entry into a different voxel, thus encoding one single traversal through this voxel. Each sample $s_K(t_i)$ carries a time stamp t_i that is attached at classification time. For non-spatial data like audio samples, only this timestamp allows the synchronization with spatial data, and therefore association with the last known location **p**. This position is provided as interleaved information together with

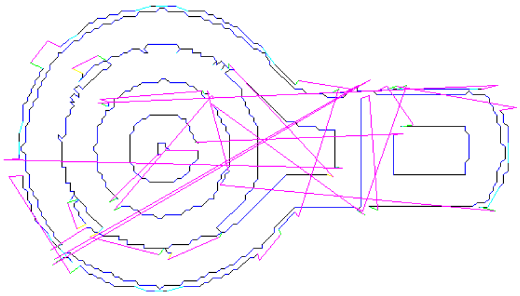

Figure 8: Perspective view on the motion path for concentric milling; originally planned path.

the force-based contact samples in the presented system, but might as well originate from other, independent sources like e.g. infrared tracking systems. By following the links to the previous and the next visited voxels with samples (the "thread"), temporal reconstruction $\mathbf{p}(t_i)$ of the followed path is achieved, which is useful for documentation purposes.

Several issues still persist with the presented approach, though. Regardless of the resolution of the grid, the inserting module has to determine the set of voxels to modify. (This question is closely related with the overall achievable resolution.) The envisioned application effectively probes the environment with a tool tip of diameter of the order d_{Fr} = 5mm. Apart from very special cases, we cannot determine the exact location or shape of the contact surface V_p between miller and environment. Therefore, when entering the samples into the grid, we cannot achieve a resolution higher than the tool tip diameter. As duplication of information should be avoided for general performance considerations, the tool tip is abstracted to a generic miller position at a reference point at the lower tip, and samples are entered only at this point, regardless of the extent of overlapping into other voxels.

Therefore, the task of querying the map basically consists of the following steps. The query function $q_M(\mathbf{p},t_q)$ has several uses which are available through appropriate parameterization. In any case, a suitable neighbourhood V_p of the position **p** which is to be queried has to be determined first, depending on effective miller radius d_{Fr}. Due to the above-mentioned fact, this neighbourhood has non-trivial shape, though we hope that further experimentation will prove detailed modeling unnecessary. Then, depending on the purpose of the map – documentation or in-time navigation – appropriate sample sublists corresponding to query time t_q have to be found within V_p. Useful query times t_q are the earliest possible time $t_q = t_{EARLY}$ with $t_q < t_i$ for all samples $s_K(t_i)$ (to retrieve the oldest available contact state information, e.g. for registration checking), or some time in the future $t_q = t_{LATE}$ with $t_q > t_i$ for all samples (to determine the most current known situation, to plan new paths). Finally, it picks a resultant contact state $C_{p,tq}$ according to task- and situation-dependent risk evaluations from the contact state histograms of those sublists.

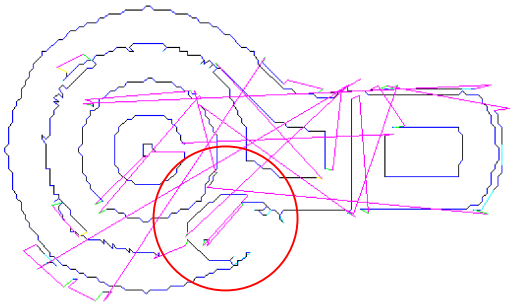

Figure 9: Perspective view on the motion path for concentric milling; with excluded subregion at lower center, after complete replanning.

VI. Navigation – Path Modification Options

One goal of map-building with local sensor data is navigation on the local map. Initially following the pre-determined path, the system iteratively builds a current view of the environment and possibly re-plans its motion. Since local and global maps show the same abstract behaviour (they allow queries for contact state $C_p = q_M(\mathbf{p},\cdot)$ at position **p**), it is easy to convert the local map into a data type accepted by the path planning algorithm [14]. Furthermore, this algorithm is able to re-plan motion on selected subregions, so modification of the path can be done on-line during process execution. Two options are *conservative modification*, i.e. keeping the planned path and skipping over critical regions, and *complete replanning*, i.e. performing path planning anew on the modified insertion profile (the original design surface from Figure 8 with newly excluded subregions, as in Figure 9).

VII. Conclusions

We have demonstrated how local navigation can be introduced into a standard robot system, based on the example of a surgical robot. Two local sensors, force and audio sampling, provide current data from the process. This data is only temporally ordered. It is then classified according to the presented system state scheme and associated with position information, so that it can be entered into a newly created spatial local map of the environment which provides an additional view independent from the pre-operative imaging data in the surgery example.

Currently, we are investigating several questions with the presented prototype of a local navigation system; those can be coarsely divided into sensor and classification problems and navigation issues.

For the force-based system state classification, according to first experiences, using estimated environment compliance (absolute force normalized with robot speed) is preferable over absolute force as basic data stream, since interference of different control algorithms is minimized. For audio classification, some system states exhibit strongly non-stationary behaviour in terms of spectral frequencies. Therefore, a classification method that tracks the state over several frames seems necessary (joint time-frequency methods like Hidden Markov Modelling or principal frequency tracking may be valid options). Furthermore, exchanging information between classifiers in a networked fashion and sensor fusion on a feature level, creating a composed feature vector, promises to enhance result quality.

Another issue is the mechanical deflection of the tool tip under load. To increase spatial precision of map-building, we will include bending in position determination.

Finally, integration of a redundant tracking system (e.g. infrared 3D navigation) will increase the safety of the envisioned application.

ACKNOWLEDGMENT

This work is a result of the project "Robot-based navigation for milling at the lateral skull base (RONAF)" of the special research cluster "Medical navigation and robotics" (SPP 1124) funded by the Deutsche Forschungsgemeinschaft (DFG), performed in cooperation with the Universitäts-HNO-Klinik (Abt. HNO-Heilkunde) in Heidelberg/Germany. Further information can be found at http://ai3.inf.uni-bayreuth.de/projects/ronaf/.

REFERENCES

[1] I. Boesnach, M. Hahn, J. Moldenhauer, Th. Beth, U. Spetzger, "Analysis of Drill Sound in Spine Surgery", *MRVN 2004 (Medical Robotics, Navigation and Visualisation)*, Remagen/Germany, March 2004.

[2] D. Engel, J. Raczkowsky, H. Wörn, "Sensor-aided Milling with a Surgical Robot System", *CARS 2002*, H.U. Lemke et al. (eds.), 2002.

[3] Ph. A. Federspil, U. W. Geisthoff, D. Henrich, P. K. Plinkert, "Development of the First Force-Controlled Robot for Otoneurosurgery", *The Laryngoscope*, Lippincott Williams & Wilkins, Inc., Philadelphia, 2003.

[4] Ph. A. Federspil et al., "Kraftgeregelter Robotervorschub verhindert Hitzeschäden beim robotergestützten Fräsen an der lateralen Schädelbasis", *CURAC 2003*, Nürnberg/Germany, 2003.

[5] Ph. A. Federspil, S. H. Tretbar, U. Geisthoff, B. Plinkert, P. K. Plinkert, "Ultrasound Based Navigation of Robotic Drilling at the Lateral Skull Base", *CARS 2003*, H.U. Lemke et al. (eds.), 2003.

[6] F. J. Harris, "On the Use of Windows for the Harmonic Analysis with the Discrete Fourier Transform", *Proceedings of the IEEE, Vol. 66, No. 1*, Jan 1978.

[7] Henrich, D.; Plinkert, P.K.; Federspil, Ph.A.; Plinkert, B., "Robot-based milling at the lateral skull base: Force-based local navigation for making implant cavitities", *VDI-Bericht 1679 – Tagungshandbuch zur Robotik 2002*, Ludwigsburg, 2002.

[8] D. Henrich and Ph. Stolka, "Principles of Navigation in Surgical Robotics", *MRVN 2004 (Medical Robotics, Navigation and Visualisation)*, Remagen/Germany, March 2004.

[9] D. Malthan et al., "Improvement of Computer- and Robot-Assisted Surgery at the Lateral Skull Base by Sensory Feedback", *MRNV 2004 (Medical Robotics, Navigation and Visualisation)*, Remagen/Germany, March 2004.

[10] R. Rojas, *"Theorie der neuronalen Netze"*, Springer Verlag Berlin Heidelberg, ISBN 3-540-56353-9, 1993.

[11] J. Rosen, B. Hannaford, C. G. Richards, M. N. Sinnan, "Markov Modeling of Minimally Invasive Surgery Based on Tool/Tissue Interaction and Force/Torque Signatures for Evaluating Surgical Skills", *IEEE Transactions on Biomedical Engineering, Vol. 48, No. 5*, May 2001.

[12] Ph. Stolka and D. Henrich, "A Hybrid Force-Following Controller for Multi-Scale Motions", *SYROCO 2003*, Wrocław/Poland, 2003.

[13] M. Waringo, Ph. Stolka, and D. Henrich, "First System for Interactive Position Planning of Implant Components", *CURAC 2003*, Nürnberg/Germany, 2003.

[14] M. Waringo, Ph. Stolka, and D. Henrich, "3-Dimensional Layered Path Planning for Anytime Milling Applications", *CURAC 2003*, Nürnberg/Germany, 2003.